

Oracle Financials[®] and Oracle Government Financials[™] Open Interfaces Manual

**RELEASE 10.7
VOLUME 1**

February 1997

ORACLE[®]

**Your Decentralized Accounting and Management
Solution**

Oracle Financials and Oracle Government Financials Open Interfaces Manual
Release 10.7 Volume 1

The part number for this volume is A36127-2. To reorder this manual, use the set part number A22516-2.

Copyright © 1997

All rights reserved. Printed in the U.S.A.

Major Contributors: Christopher Andrews, Tracy Drinkwater, Caroline Guenther

Contributors: D. Yitzik Brenman, Steve Carter, Rick Copeland, Metin Gökçen, Anne Jaeckel, Cora Kemp, Liza Lyons, Beth Mitcham, Thornton Prayer, Maria Quinlan

This software was not developed for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It is the customer's responsibility to take all appropriate measures to ensure the safe use of such applications if the programs are used for such purposes.

This software/documentation contains proprietary information of Oracle Corporation; it is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited.

If this software/documentation is delivered to a U.S. Government Agency of the Department of Defense, then it is delivered with Restricted Rights and the following legend is applicable:

Restricted Rights Legend

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of DFARS 252.227-7013, Rights in Technical Data and Computer Software (October 1988).

Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

If this software/documentation is delivered to a U.S. Government Agency not within the Department of Defense, then it is delivered with "Restricted Rights," as defined in FAR 52.227-14, Rights in Data - General, including Alternate III (June 1987).

The information in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error-free.

Oracle Alert, Oracle Financials, Pro*C, SQL*Forms, SQL*LoaderOps, SQL*Net, SQL*Plus, SQL*Report, and SQL*ReportWriter, are registered trademarks of Oracle Corporation

Oracle Alliance Program (SM), Oracle Application Object Library, Oracle Book, CASE*Generator, Oracle7, Oracle Add-In for 1-2-3, Oracle Application Display Manager, Oracle Applications, Oracle Applications Window Manager, Oracle Assets, Oracle Business Manager, Oracle Data Browser, Oracle Data Query, Oracle General Ledger, Oracle Government Financials, Oracle Human Resources, Oracle Inventory, Oracle*Mail, Oracle Manufacturing, Oracle Order Entry, ORACLE Parallel Server, Oracle Payroll, Oracle Payables, Oracle Project Accounting, Oracle Purchasing, Oracle Receivables, PL/SQL, Pro*C, and SQL*TextRetrieval are trademarks of Oracle Corporation.



Contents

VOLUME 1

Preface	Preface i Introduction to Oracle Applications ii
Chapter 1	Open Interfaces 1 – 1 Importing Information into Oracle Financials Applications 1 – 2
Chapter 2	Oracle Assets Open Interfaces 2 – 1 Importing Asset Information into Oracle Assets Using Mass Additions 2 – 2 Integrating Oracle Assets Using the Budget Interface 2 – 38 Integrating Oracle Assets Using the Production Interface 2 – 46 Integrating Oracle Assets Using the ACE Interface 2 – 53
Chapter 3	Oracle Cash Management Open Interfaces 3 – 1
Chapter 4	Oracle General Ledger Applications Open Interfaces 4 – 1 Integrating Oracle General Ledger or Oracle Government General Ledger Using Journal Import 4 – 2 Integrating Oracle General Ledger or Oracle Government General Ledger Using Budget Upload 4 – 33

Chapter 5	Oracle Inventory Open Interfaces	5 – 1
	Open Demand Interface	5 – 2
	Open Transaction Interface	5 – 22
	Open Replenishment Interface	5 – 46
	Open Item Interface	5 – 58

Chapter 6	Oracle Order Entry Open Interfaces	6 – 1
	Ship Confirm Open Interface	6 – 2
	Integrating Oracle Order Entry with Oracle Receivables	6 – 15
	Integrating Oracle Order Entry Using OrderImport	6 – 48

Chapter 7	Oracle Payables Applications Open Interfaces	7 – 1
	Integrating Payment Reconciliation Information with Oracle Payables or Oracle Government Payables	7 – 2
	Importing Invoice Information into Oracle Payables or Oracle Government Payables	7 – 20
	Integrating Purchase Order Information with Oracle Payables or Oracle Government Payables	7 – 50

Index

VOLUME 2

Chapter 8	Oracle Project Accounting Open Interfaces	8 – 1
	Importing Transaction Data into Oracle Project Accounting	8 – 2

Chapter 9	Oracle Purchasing Applications Open Interfaces	9 – 1
	Oracle Purchasing or Oracle Government Purchasing: Open Requisitions Interface	9 – 2
	Integrating Purge Information with Oracle Payables and Oracle Purchasing Applications	9 – 23

Chapter 10

Oracle Receivables Applications Open Interfaces	10 – 1
Importing Invoice Information into Oracle Receivables	
Applications Using AutoInvoice	10 – 2
Processing Bank Receipts Using Oracle Receivables	
Applications AutoLockbox	10 – 109
Integrating Customer Information Using Oracle Receivables	
Applications Customer Interface	10 – 143
Integrating Oracle Receivables Applications Tax Information	
Using Sales Tax Rate Interface	10 – 198
Implementing the Oracle Receivables Applications	
Tax Vendor Extension	10 – 219

Index



Preface

This *Oracle Financials and Oracle Government Financials Open Interfaces Manual* contains up-to-date information about integrating with other Oracle Financials and Oracle Government Financials applications and with your other systems.

Note: The open interfaces described in this manual are similar for both Release 10.7 and Release 10SC products. You can also refer to any product's Release 10SC user's guide for information about that product's open interfaces.

This chapter explains how you should use this manual, and defines the notational conventions you need to understand.

Note: This documentation includes open interfaces found in Oracle Financials Release 10.7. If you are using an earlier version of the software, please consult your support representative or the Product Update Notes for more specific information about your release.

We want you to get the most out of Oracle Financials and Oracle Government Financials.

Introduction to Oracle Applications

Oracle Applications is a complete set of tightly integrated cross-industry applications, which includes Oracle Financials, Oracle Human Resources, Oracle Manufacturing, Oracle Government Financials, and Oracle Project Accounting product families.

Oracle Applications Product Families

Technology is hastening the pace of worldwide commerce. Businesses today are larger, more geographically dispersed, and increasingly diverse. Timely information for better cost control, intelligent decision making, and improved manufacturing provides one of today's competitive advantages. Oracle's integrated families of applications, Oracle Financials, Oracle Manufacturing, Oracle Project Accounting, and Oracle Human Resources, gives you that advantage. And, Oracle Government Financials provides the same advantages for government agencies and institutions of higher education.

Our applications use the Oracle7 server, and are built using Oracle Application Object Library, our applications development tool, to bring you a standardized, integrated, full-function family of application software products.

With Oracle Applications, you can deliver on the promises of decentralized applications, distributed processing, heterogeneous computing, and connectivity.

Who Should Read This Manual

Your *Oracle Financials and Oracle Government Financials Open Interfaces Manual* is a valuable source of information about Oracle Applications integration with other Oracle Financials and Oracle Government Financials products and with your other systems. This manual can help persons responsible for implementing Oracle Applications. These people typically include:

- Project Leaders
- Systems Analysts
- Department managers
- Application Programmers

- System Programmers
- System Managers
- Database Administrators

Prerequisites to Using this Manual

In general, this manual assumes only a basic understanding of Oracle Applications, and, in some cases, a very basic understanding of your computer's operating system. However, some of the implementation decisions you make require a good understanding of how you will use your Oracle Applications products.

How This Manual is Organized

Open Interfaces

These chapters provide information about importing information using Oracle Applications open interfaces.

Notational Conventions

We use several notational conventions to make this manual easy to read and understand.

Oracle Applications Field Values

We generally use initial capitals for the names of Oracle Applications fields. For example, Application Name is the name of a field on several system administration forms.

When we ask you to enter field values into Oracle Applications forms, we precede the field value with the word "choose" or "enter".

Example Enter Oracle Application Object Library in the Application Name field.

Example Choose System Administrator from the list of responsibilities

Menu Choices

Oracle Applications menu choices begin with a backslash [\].

\ Navigate Concurrent Request

You access an Oracle Applications product menu by pressing the backslash key [\], and you can select a menu choice either by highlighting your menu choice and pressing the [Return] key or by entering the menu choice sequence, which usually corresponds to the first letter of each word.

Suggestions

Throughout this *Oracle Financials and Oracle Government Financials Open Interfaces Manual*, we provide you with helpful suggestions. These suggestions help you get the most out of Oracle Applications by taking full advantage of its powerful features and rich functionality. We highlight these suggestions with an easy-to-notice legend. Here is a sample suggestion:



Suggestion: Use a name for each depreciation type that is short enough to display fully within an 8 character field.

Warnings

Throughout this *Oracle Financials and Oracle Government Financials Open Interfaces Manual*, we provide you with warning messages. These warnings help prevent errors and save time. We highlight these warnings with an easy-to-notice legend. Here is a sample warning:



Warning: Before you can set up Oracle Purchasing to use Budgetary Control, you need to complete your Budget setup in Oracle General Ledger.

For More Information

Throughout this *Oracle Financials and Oracle Government Financials Open Interfaces Manual*, we tell you about additional, related information. We provide you with the names of the section, chapter, and manual where you can find this information. We include the name of the manual only if we reference a manual other than this one.

We highlight these references in an easy-to-notice graphic. Here's an example:



Notational Conventions

In this case, the section is Notational Conventions in this manual.

Because we designed Release 10 Oracle Applications to work together, you should refer only to other Release 10 manuals. If a manual does not say Release 10 on the front cover, it may not have the most up-to-date information about the feature you are interested in.

System Interaction

Operating system prompts and messages, Oracle Applications display messages, and commands you enter appear in this typeface:

```
$ found
```

The operating system prompt is represented by \$. Your system prompt may be different.

Filenames

Filenames appear in boldface print, as in **expdat.dmp**. Portions of a filename that may vary appear in italics, as in *filex.dat*.

Variable Input

File or device names specific to your installation appear in italics within angle brackets (< >). Be prepared to substitute the specific name you use for the generic name in the angle brackets. Do not include the angle brackets when substituting your values for these names. Optional arguments appear within square brackets ([]).

```
$ cd /<appl_top>/<version>/bin
```

Example: (where *<appl_top>* = finance and *<version>* = account)

```
$ cd /finance/account/bin
```

ORACLE Names

The names of ORACLE user IDs, tables, views, synonyms, and columns are capitalized.

For example, we capitalize the ORACLE user ID that accesses your Oracle Financials demonstration database: MODEL10/FND.

Do Not Use Database Tools to Modify Oracle Applications Data

Oracle provides powerful tools you can use to insert, update, and delete information in an Oracle7 database. But, if you use Oracle tools like SQL*Plus or Oracle Data Browser to modify Oracle Applications data, you risk destroying the integrity of your data and you lose the ability to audit changes to your data.

Because Oracle Applications tables are interrelated, any change you make using an Oracle Applications form can update many tables at once. But when you modify Oracle Applications data using anything other than Oracle Applications forms, you may change a row in one table without making corresponding changes in related tables. If your tables get out of synchronization with each other, you risk retrieving erroneous information and you risk unpredictable results throughout Oracle Applications.

When you use Oracle Applications forms to modify your data, Oracle Applications automatically checks that your changes are valid. Oracle Applications also keeps track of who changes information. But, if you enter information into database tables using database tools, you may store invalid information. You also lose the ability to track who has changed your information because SQL*Plus and other database tools do not keep a record of changes.

Consequently, we STRONGLY RECOMMEND that you never use SQL*Plus, Oracle Data Browser, database triggers, or any other tool to modify Oracle Applications tables, unless we tell you to do so in our installation, implementation, or open interface manuals.

Other Information Sources

There are additional information sources which you can use to increase your knowledge and understanding of Oracle Applications products. We want to make Oracle Applications easy for you and your staff to understand.

Oracle Financials and Oracle Government Financials Changes

This book provides a complete list of the features and enhancements that are new with this release of Oracle Financials and Oracle Government Financials.

Oracle Financials and Oracle Government Financials Implementation Manual

This book guides you step-by-step through your set up of the Oracle Financials or Oracle Government Financials products. This manual also provides information about implementing multiple sets of books.

Oracle Applications User's Guide

This book tells you everything you need to know about entering data, querying, running reports, and other basic features of every Oracle Applications product.

Oracle Applications Demonstration User's Guide

This manual documents the functional storyline and product flows for Global Computers, a fictional manufacturer of personal computers products and services. As well as including product overviews, the book contains detailed discussions and examples, across each of the major product flows. Tables, illustrations, and charts summarize key flows and data elements.

Oracle Applications Reference Manuals

These manuals are available for each Oracle Applications product. They provide detailed information on all product functions, forms, and reports, including essays showing you how to customize the product to your organization and how to use the full power of the product.

Oracle Applications Flexfields Manual

This manual provides flexfields planning, setup and reference information for the Oracle Applications implementation team, as well as for users responsible for the ongoing maintenance of Oracle Applications product data. This manual also provides information on creating custom reports on flexfields data.

Use this manual as a companion to your *Oracle Financials and Oracle Government Financials Implementation Manual* and your Oracle Applications product reference manuals.

Oracle Applications Technical Reference Manuals

These manuals contain database diagrams and a detailed description of database tables, forms, reports, and programs. This information helps you convert data from your existing applications, integrate your Oracle

Applications products with other non-Oracle applications, and write custom reports.

You can order a Technical Reference Manual for any product you have licensed.

Oracle Applications Installation Manual

This manual provides information you need to successfully install Oracle Financials, Oracle Government Financials, Oracle Manufacturing, or Oracle Human Resources in your specific hardware and operating system software environment.

Oracle Applications Upgrade Preparation Manual

This manual explains how to prepare your Oracle Applications products for an upgrade. It also contains information on finishing the upgrade procedure for each product. Refer to this manual and the *Oracle Applications Installation Manual* when you plan to upgrade your products.

Oracle Applications Database Changes Manual

This manual lists the changes to the database structure of Oracle Applications between Release 9 and Release 10. It does not provide full information about each product's database structure, however. It is intended to alert you to the changes that may affect customizations.

Oracle Applications System Administration Reference Manual

This manual provides planning and reference information for the Oracle Applications System Administrator.

Oracle Applications Message Reference Manual

The *Oracle Applications Message Reference Manual* contains an explanation of each Oracle Applications message. (Oracle Applications messages begin with the prefix "APP-".) It also provides information on the causes of each message and the actions you take if you get a message.

You can read this manual only in Oracle Book electronic format—we do not publish a paper version of this manual.

Of course, you can always see the cause and action for any message you encounter using an Oracle Applications Help function.



Viewing Message Explanations
(*Oracle Applications User's Guide*)

Training

We offer a complete set of formal training courses to help you and your staff master Oracle Applications and quickly reach full productivity. We organize these courses into functional learning paths, so you take only those courses appropriate to your job or area of responsibility.

And, you have a choice of educational environments. You can attend courses at any one of our many Education Centers, or you can arrange for our trainers to teach at your facility. In addition, Oracle consultants can tailor standard courses or develop custom courses to meet your needs. For example, you may want to use your organization structure, terminology, and data as examples in a customized training session delivered at your own facility.

Support

From on-site support to central support, our team of experienced professionals continually provides you with whatever help and information you need to keep Oracle Applications working for you. This team includes your Technical Representative, Account Manager, and Oracle's large staff of consultants and support specialists with expertise in your business area, Oracle7 database management, and your particular hardware and software environment.

About Oracle

Oracle develops and markets an integrated line of software products for database management, applications development, decision support, and office automation, as well as a complete family of financial, manufacturing, project accounting, distribution, and human resource applications (including Oracle Applications).

Oracle products are available for mainframes, minicomputers, and personal computers, allowing organizations to integrate different computers, different operating systems, different networks, and even different database management systems, into a single, unified computing and information resource.

Oracle offers its products, along with related consulting, education, and support services, in most countries around the world. Oracle is the world's largest supplier of database management software and services.

Thank You

Thank you for using Oracle Applications and this implementation manual.

As an Oracle Applications user, we appreciate your comments and feedback. At the end of this manual is a Reader's Comment Form which you can use to explain what you like or dislike about Oracle Applications or manual. Mail your comments to the following address:

Oracle Applications Documentation Manager
Oracle Corporation
500 Oracle Parkway
Redwood City, CA 94065

Open Interfaces

This chapter describes the open interfaces we provide with Oracle Financials applications products. We explain how you can integrate your products with non-Oracle systems, including:

- Integrating Oracle Financials applications with your proprietary or custom applications
- Importing historical data from your previous application system into your Oracle Financials applications
- Importing information from various financial information providers
- Importing information from spreadsheets

This chapter also contains:

- Additional information about the integration between certain Oracle Financials products

Oracle Financials open interfaces are powerful, flexible tools that allow you to capture data from your own applications, define necessary format conversions, and direct data to your Oracle Financials products, usually without programming.

Importing Information into Oracle Financials Applications

Using Oracle Financials and Oracle Government Financials import programs, you can integrate new or existing applications such as payroll, accounts receivable, accounts payable, and fixed assets into your Oracle Financials applications. You can import accounting data from your feeder systems. You can also import historical data from your previous accounting or management systems.

Oracle Financials and Oracle Government Financials provide several open interfaces to let you import data from your proprietary or custom applications, including:

- Adjusted Current Earnings (ACE) Information
- Bank Receipts (AutoLockbox)
- Capital Budgets
- Customer Information
- Demand Requests
- Financial Budgets
- Inventory Transactions
- Invoices for both Payables and Receivables Applications
- Journal Entries
- Mass Additions
- Payment Reconciliations
- Production Information
- Purchase Order Matching
- Replenishment Requests
- Requisitions
- Sales Orders
- Tax Information

Oracle Financials and Oracle Government Financials also contain several interfaces that transfer data in between Oracle Applications. Some of these are also documented in detail in this chapter for your reference, including:

- Integrating Purge Information with Oracle Payables and Oracle Purchasing Applications
- Oracle Order Entry's Receivables Interface

Basic Importing Needs

Oracle Financials import programs provide you with the features you need to satisfy the following basic integration needs:

- Import information in the easiest way possible, and automatically convert the valid data you import into data that is meaningful to you and your organization.
- Import information from a variety of environments, including your own and other accounting systems.
- Import historical data from your previous accounting, sales order or other management systems to keep your records consistent and up-to-date.
- Review the results of your import run. Identify which data has been successfully imported, and any errors which may have occurred during the import process.
- Correct invalid data online that an import program was unable to accept.

In addition, several Oracle Financials import programs satisfy the following additional needs:

- Validate the integrity of any data before introducing it into your Oracle Financials application
- Choose to optionally archive your source data each time you run an import program

Definitions

Feeder Program

A custom program you use to import your detail accounting transactions from an external or feeder system into your Oracle Financials application. The type of feeder program you write depends on the environment from which you are importing data.

ORACLE Tables

Rows and columns by which data is organized in the ORACLE Relational Database Management System. Categories of information are listed across the top of each table, while individual cases are listed down the left side. In this form you can readily visualize, understand

and use the information. Oracle Financials products use ORACLE tables to store the information you need to run your business.

Table Columns

ORACLE tables consist of columns. Each column contains one kind of information. We use the following convention for naming table columns: TABLE_NAME.COLUMN_NAME.

Column Reference

Column in a table which references information stored in a column in another table. When columns in different tables contain the same type of information, the common information permits entries in the two tables to be combined or related to each other.

Column Types

Each column in a table can hold one *type* of value. The most common types are:

Char	A char column contains a sequence of up to 240 letters, numbers, punctuation marks and special characters like +, -, % and \$
Number	A number column contains a number consisting of digits, a sign and a decimal point
Date	A date column contains a date and time of day

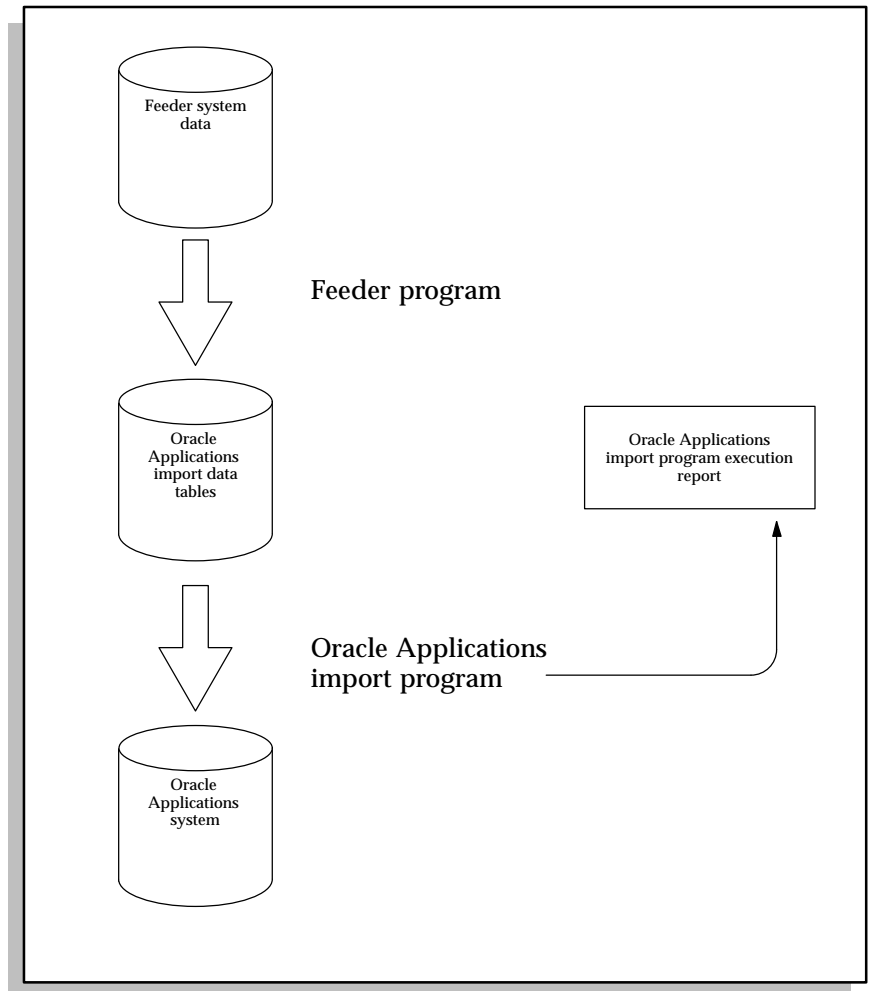
Not Null Column

A column in which you must enter information. In other words, a column in which the value may not be null.

Importing Data From Your Feeder System

Oracle Applications can receive data from an *import program* which your on-site MIS personnel can develop for you. Or, you may wish to use an Oracle consultant. The goal of your import program is to convert data from your feeder system into a standard data format that your Oracle applications can read and then convert for further modification or processing in your Oracle application.

Figure 1 - 1
Importing Data into an
OracleApplications system



Writing a Feeder Program

The type of environment from which you want to import data determines the type of feeder program you need to write. For example, you can use SQL*Loader to write an import program to feed data from a non-Oracle system. Or, you can write a feeder program to import historical data from your previous accounting system. Regardless of the type of feeder program you write, the output should be in standard data format that an Oracle Applications import program can use to convert your import data into your Oracle Applications system.

Choosing a Feeder Program

You need to choose a tool for writing a feeder program to extract data from your existing application system's printed reports, flat file, relational database, or other repository of application information. Using a feeder program you write, you populate an Oracle Applications import table with the information you want to introduce to your Oracle Financials system.

SQL*Loader is a powerful and easy-to-use tool you can to write a feeder program. SQL*Loader lets you map elements of a regularly formatted file, such as a listing or flat file, and specify which columns of which tables to populate. Chances are, SQL*Loader is a powerful enough tool to use for your feeder program.

If you need to do more complex computations to extract or rearrange information from your feeder system, you can choose from a variety of programming languages to populate an import table.

For example, you can use any of the Oracle-enhanced programming languages, such as Pro*C, Pro*COBOL, or Pro*FORTRAN, as well as SQL*Report.

Oracle Assets Open Interfaces

This chapter contains information about the following Oracle Assets open interfaces:

- Importing Asset Information into Oracle Assets Using Mass Additions
- Integrating Oracle Assets Using the Budget Interface
- Integrating Oracle Assets Using the Production Interface
- Integrating Oracle Assets Using the ACE Interface

Importing Asset Information into Oracle Assets

Using Mass Additions

Oracle Assets lets you create assets automatically from information in any other system using mass additions. Oracle Assets is already integrated with Oracle Payables, but you can easily integrate it with other systems. Or, you can use the mass additions process to convert your assets from an outside system to Oracle Assets.

Using Mass Additions To Create Assets From Oracle Payables

The Create Mass Additions program creates mass additions from invoice information in Oracle Payables. The concurrent process places the new mass additions in a holding area (the table FA_MASS_ADDITIONS) that is separate from the main Oracle Assets tables, so that you can review and approve the mass additions before they become asset additions.



Using Mass Additions
(*Oracle Assets Reference Manual*)

Using Mass Additions To Create Asset Additions From Another Payables System

To integrate Oracle Assets with another system, develop your own program to add mass additions to the FA_MASS_ADDITIONS table. Also, you may want to add another form to the Oracle Assets menu to run the concurrent process. A description of the columns in the FA_MASS_ADDITIONS table is included later in this essay.



Importing Your Asset Information (See page 2 – 19)

Using Mass Additions To Convert From Other Systems

Oracle Assets lets you convert from your previous asset system using mass additions. Instead of loading your asset information into multiple Oracle Assets tables, load your information into the FA_MASS_ADDITIONS table and use the mass additions process to simplify your work.

This essay provides you with a specific recommended plan of action you use to convert outside asset information into Oracle Assets tables. It is organized in a plan format which you should follow, step by step, to complete your conversion. Plan your conversion carefully and thoroughly, since you cannot undo it.

Terms

The following terms are used in this essay. You may wish to refer to these definitions as you read if you are not familiar with them.

SQL*Loader

The name of an Oracle7 Server Utility which accepts operating system files as input to Oracle7 tables. Oracle Assets conversions often require loading information from foreign databases or file systems. If the existing data is not already in an Oracle7 table, you can use SQL*Loader to import it. SQL*Loader can import data from a wide variety of formats into multiple Oracle7 tables.



Additional Information: *Oracle7 Server Utilities User's Guide*

FA_MASS_ADDITIONS Interface Table

The Oracle Assets table into which you load your data. This table is normally used to link Oracle Assets with Oracle Payables. You use this table to store and then post your asset information to Oracle Assets, using the Mass Additions process. After you load asset information into this table, you can examine each line in the Prepare Mass Additions form.

Create Mass Additions

The Oracle Payables program you use to load the FA_MASS_ADDITIONS table if you are using Oracle Payables. Create Mass Additions selects invoice line items which could become assets. You can review them in the Prepare Mass Additions form.



Create Mass Additions for Oracle Assets
(*Oracle Payables Reference Manual*)

Prepare Mass Additions

The Oracle Assets form you use to review your mass addition lines to determine if they should become assets and enter additional information. Mass Additions lines in the Prepare Mass Additions form are not yet assets. Mass addition lines do not become assets until you put them into the POST queue and run Mass Additions Post.



Prepare Mass Additions
(*Oracle Assets Reference Manual*)

Mass Additions Post

The Oracle Assets program you use to post assets to Oracle Assets tables. You run Mass Additions Post from the Send Mass Additions to Oracle Assets form.



Send Mass Additions to Oracle Assets
(*Oracle Assets Reference Manual*)

Why Use Mass Additions to Import Your Asset Data

The Mass Additions feature of Oracle Assets is normally used to import asset information from Oracle Payables. Mass Additions automatically populates the many Oracle Assets tables from the relatively simple FA_MASS_ADDITIONS table. By placing your data in this table, you can use the power of the Mass Additions Post program to perform the bulk of your import.

Mass Additions has three main components:

- **Create:** finds potential new assets in Oracle Payables and brings them into the FA_MASS_ADDITIONS table
- **Prepare:** allows you to review potential new assets and enter additional information
- **Post:** creates assets by importing asset information from the FA_MASS_ADDITIONS table into several Oracle Assets tables

To import asset information from another payables system, load the FA_MASS_ADDITIONS table and then use Prepare and Post to add your assets to Oracle Assets.

To convert assets from another assets system, use only the Post component to move the asset information you store in the FA_MASS_ADDITIONS table into Oracle Assets.

Major Steps in Importing Asset Information

Your asset information import consists of five major steps. Follow these steps to ensure that your import is successful.

Importing Checklist

Complete these steps to import your asset data into Oracle Assets. The remaining sections of this chapter explain each step in detail.

- ☐ 1. Plan your import
- ☐ 2. Define your Oracle Assets setup information and perform the Oracle Assets setup process
- ☐ 3. Load your asset data into an interim table or file you define
- ☐ 4. Import or load your asset data from the interim table or file into FA_MASS_ADDITIONS
- ☐ 5. Reconcile your Oracle Assets data with your old asset information

Importing Steps

Step 1 **Plan Your Import**

During this phase of your import, you plan each step of the import process. Take care to plan enough time to complete all the required steps. Prepare a complete step-by-step procedure to follow, with distinct checkpoints.

Step 2 **Define Your Oracle Assets Setup Information And Perform The Oracle Assets Setup Process**

During this phase, make important business decisions about how to define Oracle Assets and then actually perform the setup process.

Step 3 **Load Your Asset Data Into An Interim Table Or File You Define**

During this phase, define an interim table that closely matches the structure of the existing asset data. Then load the data into it using SQL*Loader if required.

Step 4 Load Your Asset Data Into FA_MASS_ADDITIONS

During this phase, use SQL*Plus to move your asset data into the FA_MASS_ADDITIONS table. Run the Mass Additions Status Report and the Unposted Mass Additions Report to check your data. Then prepare the assets using Prepare Mass Additions if necessary and post them using Mass Additions Post.

Step 5 Reconcile Your Oracle Assets Data With Your Old Asset Information

During this phase, compare your converted asset data against your old asset records to ensure that they match.

Planning Your Import

Use the information in this section to accurately schedule your Oracle Assets import.

Planning Checklist

Complete the following steps to prepare for your import:

- ☐ 1. Decide when to begin your import project
- ☐ 2. Know your ability to make the important decisions you need to complete your import
- ☐ 3. Schedule your equipment to make sure you have adequate computer resources available when you need them
- ☐ 4. Know the status of your Oracle Assets installation
- ☐ 5. Know in what form the other system stores its data, and what tools are available to convert that data into a form readable by SQL*Loader, if necessary
- ☐ 6. Know the number of assets, categories, locations, and exceptions in your data
- ☐ 7. Define a import schedule
- ☐ 8. Reach agreement with all parties on the schedule
- ☐ 9. Complete schedule documentation

Planning Steps

Step 1 Decide When To Begin Your Import Project

Choose a time when you are able to devote enough time to complete the entire import process.

Step 2 Know Your Ability To Make The Important Decisions You Need To Complete Your Import

Determine who, if not you, can make the decisions required to complete the definition phase found in the next section. Ensure that you have the correct person available to make these decisions.

Step 3 Schedule Your Equipment To Make Sure You Have Adequate Computer Resources Available

Schedule and confirm the computer resources you need, well in advance. Specifically, make sure that you have:

- Disk space sufficient to handle multiple copies of your asset data
- Access to a terminal
- Access to a media drive (if required)
- Sufficient CPU and memory to handle the database operations you perform as part of the import
- Access to operating system documentation

Step 4 Know The Status Of Your Oracle Assets Installation

Oracle Assets must be installed and confirmed before your import. You use Oracle Assets tables and programs during your import. If you will be doing an install, consult your Oracle Applications Install Manual for complete installation instructions. Determine if you are using Oracle General Ledger and if it is installed and set up.

Step 5 Know In What Form The Other System Stores Its Data, And What Tools Are Available To Convert That Data

Review the current form of the asset data. You need to know what it will take to convert the existing asset data into a form that SQL*Loader can read.

Step 6 Know The Number Of Assets, Categories, Locations, And Exceptions In Your Data

You need these numbers to estimate the amount of time required for the import.

Step 7 Define An Import Schedule And Procedure

Estimate how long it will take you to complete your import. Base your schedule on the information you gained in the preceding steps. Be sure to budget enough time, including a buffer in case you have difficulty during the import.

Step 8 Make Sure All Parties Agree On The Schedule

All parties need to understand and agree to the schedule you define.

Step 9 Complete Schedule Documentation

Complete your schedule documentation which clearly states the start and end dates for each major step, and the amount of time for each detail step.

Defining Oracle Assets

Use this section to make the decisions you need to successfully implement Oracle Assets and convert existing asset data. Once all the decisions are made, follow the setup procedure for Oracle Assets to implement your decisions. Defining Oracle Assets is the most important phase of a conversion, so take sufficient care to make decisions that you can live with in the long term.



Overview of Setting Up
(*Oracle Assets Reference Manual*)

Defining Checklist

Complete the following steps to define Oracle Assets before you begin importing your asset data:

- ☐ 1. Define your Accounting Flexfield structure
- ☐ 2. Define your Location Flexfield structure
- ☐ 3. Define your Category Flexfield structure
- ☐ 4. Define your Asset Key Flexfield structure
- ☐ 5. Define locations, cost centers, asset key flexfield combinations and vendors
- ☐ 6. Define when to perform the initial depreciation run
- ☐ 7. Define how the assets will be numbered
- ☐ 8. Define what depreciation methods apply to each asset in each book
- ☐ 9. Define your categories
- ☐ 10. Complete the Oracle Assets Setup procedure
- ☐ 11. Define your asset key values

Defining Steps

Step 1 Define Your Accounting Flexfield Structure

Most sites do not run Oracle Assets as a stand-alone system, so an accounting flexfield may already exist for your Oracle General Ledger. If you plan to run Oracle Assets as a stand-alone system, you need to define the Accounting Flexfield.

Step 2 Define Your Location Flexfield

Most companies plan to do business internationally, if they do not already, so the location flexfield must begin with the country. State, possibly county, city, and site are the typical segments of a location flexfield. Many companies find it useful to pinpoint the exact building and room for some assets, for example, for barcoding. Add these segments if needed.

Step 3 Define Your Category Flexfield

Define your category flexfield. Most companies prefer to set up categories which match their chart of accounts. Each chart account defines a major category. Most often, the first segment, or major category corresponds to the asset accounts in the company's chart of accounts. Define at least one subcategory segment to allow for distinctions within a major category. You can define up to seven segments if required.

You probably want to set up no more than 3 category segments due to maintenance issues, and limited reporting space on Oracle Assets reports.

Potential uses for a subcategory segment include such information as personal/real, capitalized/expensed, owned/leased, project numbers, foreign, and luxury items.

Step 4 Define Your Asset Key Flexfield

A company may have a system for grouping similar assets. Many companies find it useful to group assets associated with a specific project, department, or location. For example, you can use the asset key flexfield to track your construction-in-process assets. Define an asset key flexfield that describes asset groups in your organization.

If you do not choose to track assets using the asset key, define a one segment asset key flexfield without validation. Then, when you

navigate to the asset key flexfield on a form, the flexfield window does not open and you can return through the field. You can define up to ten segments for the asset key flexfield if required.

Step 5 Define Default Locations, Cost Centers, Asset Key Flexfield Combination And Vendors

If you track this information electronically, set up each location, expense code combination, asset key flexfield combination, and vendor before import. If you do not already track this information, define default values. The defaults you define are used in place of actual values until the actual value is known.

1. Default Location

Many companies do not currently track asset location. If you do not, then set up a default location to be used in the import. Once the assets are imported, you can locate the assets by performing an asset inventory and then transfer the assets from the default location to the actual location. If you already track location, then set up each location before import.

2. Default Cost Center

Some companies do not track assets on a cost center level. If you do not, then set up a default cost center for your incoming assets. To define the default cost center, create a code combination in each depreciation expense account for the default cost center. When you start tracking assets by cost center, you can transfer each asset from the default cost center the actual one. If you already track assets by cost center, create each expense account code combination.

3. Default Asset Key Flexfield Combination

Some companies do not use the asset key flexfield. If you do not, then define a default combination to be used for incoming assets. Give the default combination an 'inactive date' so you do not use the combination accidentally at a future time. If you start using the asset key flexfield, you can change the combination from the default combination to the actual one. If you already use the asset key flexfield, create each asset key flexfield combination.

4. Default Vendors

Many companies do not track vendors on an asset level. If you do not, then define a default vendor to be used for incoming assets. Give the default vendor an 'inactive date' so you do not use the vendor name accidentally at a future time. When you start tracking asset vendors, you can change the vendor from the default

vendor to the actual one. If you already track asset vendors, then enter all vendors for all incoming assets. If you use Oracle Payables or Oracle Purchasing, your vendors are already defined there and Oracle Assets shares the information.

Step 6 Define When To Perform The Initial Depreciation Run

Typically, the best time for the initial depreciation run after a conversion is the end of the fiscal year just before the current one. This year provides accumulated depreciation numbers that you can reconcile with your company's financial statements. It also ensures that the year-to-date numbers on your reports are correct during the current fiscal year.

Decide whether to enter assets with depreciation reserve, or enter them without reserve and let Oracle Assets calculate the asset reserve for you. If you do not enter reserve, Oracle Assets calculates the depreciation reserve and revaluation reserve if necessary the first time you run depreciation. If you enter your assets with reserve, Oracle Assets does not recalculate it, unless you make an adjustment to that asset. You must enter the correct reserve.

Year-to-date numbers are important because the Depreciation program expenses all the catchup depreciation to the period in which it is run. If you enter your assets without depreciation reserve, the first time you run depreciation, the depreciation expense is equal to the accumulated depreciation. The first period absorbs that one-time expense and, since the result from that period is not be posted to the general ledger, the general ledger is not affected and your year-to-date numbers in the next year are correct in Oracle Assets.

When you want to use the values for accumulated depreciation from the old assets system, you should start with the first period of the current fiscal year. Otherwise, your year-to-date values will not include the year-to-date depreciation you specify in the FA_MASS_ADDITIONS table.

You determine the initial depreciation period when you set up your books using the Book Controls form. Enter the period of your initial depreciation period in the Current Period Name field.



Book Controls
(Oracle Assets Reference Manual)

Step 7 Define Your Asset Numbering Scheme

Determine if you want to use your own asset numbering system, or if you want to use Oracle Assets automatic numbering system. In either case you must choose a starting value for automatic numbering in the System Controls form. Even if you do not use automatic numbering for the conversion, Oracle Assets uses the value internally as the ASSET_ID, so you must still choose the starting value carefully. You must use a value that is large enough to leave sufficient asset numbers unclaimed by automatic numbering. However, you cannot use asset numbers larger than 2,000,000,000.

If you do use automatic numbering, enter a value on the System Controls form that is the starting value you want to use plus the number of assets you are converting. By entering a value larger than you use for any conversion asset, you ensure that Oracle Assets does not try to assign an existing asset ASSET_ID to a newly added asset after the conversion. Using specific ranges of numbers for groups of assets makes it easier to keep track of related items.



System Controls
(*Oracle Assets Reference Manual*)

Step 8 Define Your Depreciation Methods For All Assets In All Books

Determine the depreciation methods for all assets in all books. Oracle Assets seeds most depreciation methods, but you must enter any customized method. Decide what depreciation method, prorate convention, and other depreciation rules will be used for each asset in each book. You must set up the depreciation methods and rates, depreciation ceilings, investment tax credit rates, prorate conventions, and price indexes you will need before you can define your categories.

Step 9 Define Your Asset Categories

Use category names that match the corresponding chart asset account. Define subcategories so that all the assets in a subcategory have the same depreciation method, prorate convention, and other depreciation rules. For assets in your tax books that you acquired under different, past tax laws, you can set up the asset category with different depreciation rule defaults for different date placed in service ranges. The asset category and date placed in service determine the depreciation rule defaults for an asset.

You need to set up categories because the Mass Additions Posting program gets depreciation method information for each asset from the defaults defined in the Asset Categories form.



Asset Categories
(*Oracle Assets Reference Manual*)

If you have only a few assets that use a particular depreciation method, you do not necessarily need to define a separate category for them. Instead, you can place them in another category and then manually change the depreciation information using the Depreciation Books form before you run depreciation for the first time. Note that you should place these assets in a compatible category because the asset and reserve accounts are determined by your category choice, and the capitalize flag and the category type cannot be changed on the Depreciation Books form.



Depreciation Books
(*Oracle Assets Reference Manual*)

For each book, create a plan to give all assets the correct depreciation information. Most assets should receive the correct information from the category books values. You can change the depreciation information for individual exceptions. Create your categories to minimize the number of individual transactions.

Try to keep the major category names short, ideally less than 20 characters, since Oracle Assets reports print only a limited number of characters for the category.

Step 10 Define Your Oracle Assets Installation

Follow the standard setup procedure. With the information you obtained in the previous steps, you can complete the setup procedure and be ready to add assets.



Overview of Setting Up
(*Oracle Assets Reference Manual*)

Step 11 Define Your Asset Key Values

Define values for your Asset Key flexfield that describe the different groups of assets within the company. Many companies find it useful to group assets associated with a specific project, department, or location. You can use the asset key flexfield to track your construction-in-process assets.

Loading Your Asset Data

This section describes how to define, load, and confirm your interim asset table. In many cases you have to load asset information into the Oracle7 tables from a non-Oracle file system. This section shows you how to use SQL*Loader to import your information.

Loading Checklist

Complete the following steps to load your asset data:

- ☐ 1. Define your interim table in the Oracle database
- ☐ 2. Load your interim table
- ☐ 3. Compare record counts and check SQL*Loader files
- ☐ 4. Spot check the interim table

Loading Steps

Step 1 **Define Your Interim Table In The Oracle Database**

Use a single interim table if possible. You can use multiple tables if the data exists in multiple tables or files in the old asset system. In either case, you must eventually place the data in a single table, the FA_MASS_ADDITIONS table.

If you wish, you may load data directly into FA_MASS_ADDITIONS, but it is more difficult due to the complexity of the table.

Step 2 **Load Your Interim Table (Using SQL*Loader If Asset Data Is External)**

Use SQL*Loader to import information from outside your Oracle database. SQL*Loader accepts a number of input file formats and loads your old asset data into your interim table.

If the data already resides within an Oracle database, there is no need to use SQL*Loader. Simply consolidate the asset information in your interim table using SQL*Plus or import, and go directly to Step 3.

Follow these steps if you plan to use SQL*Loader:

1. Get the Asset Information in Text Form

Most database or file systems can output data in text form. Usually you can generate a variable or fixed format data file containing comma or space delimiters from the existing system. If you can't find a way to produce clean text data, try generating a report to disk, using a text editor to format your data. Another option is to have SQL*Loader eliminate unnecessary information during its run. If there is a large volume of information, or if the information is difficult to get in a loadable format, you can write your own import program. Construct your program to generate a SQL*Loader readable text file.

2. Create the SQL*Loader Control File

In addition to the actual data text file, you must write a SQL*Loader control file. The control file tells SQL*Loader how to import the data into your interim table. Be sure to specify a discard file if you are planning to use SQL*Loader to filter your data.

3. Run SQL*Loader to Import Your Asset Data

Once you have created your asset data file and SQL*Loader control file, run SQL*Loader to import your data. SQL*Loader produces a log file with statistics about the import, a bad file containing records that could not be imported due to errors, and a discard file containing all the records which were filtered out of the import by commands placed in the control file.



Additional Information: *Oracle7 Server Utilities User's Guide*

Step 3 **Compare Record Counts And Check The SQL*Loader Files**

Check the number of rows in the interim table against the number of records in your original asset data file or table to ensure that all asset records are imported.

The log file shows if records were rejected during the load, and the bad file shows which records were rejected. Fix and re-import the records in the bad file.

Step 4 **Spot Check Interim Table**

Check several records throughout the interim table and compare them to the corresponding records in the original asset data file or table.

Look for missing or invalid data. This step ensures that your data was imported into the correct columns and that all columns were imported.

Importing Your Asset Information

Once you have set up Oracle Assets, you are ready to import your asset information.

If you are importing asset information from another payables system, load the FA_MASS_ADDITIONS table directly. Then use Prepare Mass Additions and Mass Additions Post to add your assets to Oracle Assets.

If you are converting asset information from another assets system, use SQL*Plus to move the asset information from the interim table into the FA_MASS_ADDITIONS table. After you load and confirm the FA_MASS_ADDITIONS table, run Mass Additions Post to post your assets to Oracle Assets.

Note: If you are converting units of production assets from another system, you cannot load them with accumulated depreciation. Instead, please add the asset with zero accumulated depreciation, and then provide the life-to-date production amount for the current period. Oracle Applications then uses this production amount to calculate the catchup depreciation when you run depreciation.

Note: If you want to load a large number of units of production assets, please use the production interface instead of the mass additions interface.



Integrating Oracle Assets Using the Production Interface
(See page 2 – 46)

Importing Checklist

Complete the following steps to import your assets:

- ☐ 1. Examine and understand the FA_MASS_ADDITIONS table layout
- ☐ 2. Load asset information into the FA_MASS_ADDITIONS table
- ☐ 3. Load expense code combination ids
- ☐ 4. Load category ids
- ☐ 5. Load location ids
- ☐ 6. Load vendor information

- ☐ 7. Run Mass Additions Prepare, if necessary, and Mass Additions Post
- ☐ 8. Fix exceptions

Importing Steps

Step 1 Understand The FA_MASS_ADDITIONS Table Layout

The database definition of the FA_MASS_ADDITIONS table does not require that you provide values for any columns, but in order for the Mass Additions Posting program to work properly, you must follow the rules in the following list of column descriptions. The Mass Additions Posting program uses some of the columns in the FA_MASS_ADDITIONS table, so these columns are marked NULL. Do not import your data into columns marked NULL. You must fill columns marked REQUIRED before you run Mass Additions Post.

You can either load the column directly from your other system, or you can fill in some values in the Prepare Mass Additions form before you post. Columns marked OPTIONAL are for optional asset information that you can track if you want. VARCHAR2 columns are case sensitive.

For columns marked PREP you can either import information into the column directly or enter it in the Prepare Mass Additions form before you post.

FA_MASS_ADDITIONS Columns

<u>ACCOUNTING_DATE</u>	<u>DATE</u>	<u>REQUIRED</u>
------------------------	-------------	-----------------

Use this column for the accounting date of the invoice. The Unposted Mass Additions Report reports on this date.

<u>ADD_TO_ASSET_ID</u>	<u>NUMBER(15)</u>	<u>NULL</u>
------------------------	-------------------	-------------

Do not use this column.

<u>AMORTIZE_FLAG</u>	<u>VARCHAR2(3)</u>	<u>NULL</u>
----------------------	--------------------	-------------

Do not use this column.

AP_DISTRIBUTION_		
LINE_NUMBER	NUMBER(15)	OPTIONAL

Use this column for the distribution line number. The value must be a valid in the AP_DISTRIBUTION_LINE_NUMBER column on the table AP_INVOICE_DISTRIBUTIONS, and tied to the INVOICE_ID given above.

ASSET_CATEGORY_ID	NUMBER(15)	REQUIRED PREP
--------------------------	-------------------	----------------------

Use this column for the category id that corresponds with the asset category of the asset. See Step 4 for more information.

ASSET_KEY_CCID	NUMBER(15)	OPTIONAL PREP
-----------------------	-------------------	----------------------

Use this column for the code combination id that corresponds to the asset key. Use an enabled value from the CODE_COMBINATION_ID column of the FA_ASSET_KEYWORDS table.

ASSET_NUMBER	VARCHAR2(15)	OPTIONAL PREP
---------------------	---------------------	----------------------

This column is a unique external identifier for assets. If you enter a value in this column it is used for asset numbering. The numbers you supply must be unique. It identifies the asset in Oracle Assets forms and reports. If you leave this column blank Oracle Assets assigns an asset number using automatic numbering, in which case the asset number is the same as the asset id.

ASSET_TYPE	VARCHAR2(11)	REQUIRED PREP
-------------------	---------------------	----------------------

Use this column for the asset type of the asset. Enter one of the following values:

CAPITALIZED	For your capitalized assets
CIP	For your construction-in-process ('CIP') assets.
EXPENSED	For your expensed assets.

ASSIGNED_TO	NUMBER(15)	OPTIONAL PREP
--------------------	-------------------	----------------------

Use this column for the employee id of the employee responsible for this asset. Use a value from the EMPLOYEE_ID column in FA_EMPLOYEES. Note that EMPLOYEE_ID is the unique internal identifier, and not the same as the external identifier EMPLOYEE_NUMBER.

ATTRIBUTE_		
CATEGORY_CODE	VARCHAR2(30)	NULL

Do not use this column.

ATTRIBUTE1		
through ATTRIBUTE30	VARCHAR2(150)	NULL

Do not use these columns.

BEGINNING_NBV	NUMBER	OPTIONAL
----------------------	---------------	-----------------

Do not use this column.

BOOK_TYPE_CODE	VARCHAR2(15)	REQUIRED
-----------------------	---------------------	-----------------

Use this column for the book to which you want to assign the asset. You must choose a book that you have set up in the Book Controls form, and the book class must be CORPORATE. Values in this column must be in upper case.

CREATE_BATCH_DATE	DATE	OPTIONAL
--------------------------	-------------	-----------------

Use this column for the date you load this asset into the FA_MASS_ADDITIONS table.

CREATE_BATCH_ID	NUMBER(15)	OPTIONAL
------------------------	-------------------	-----------------

Use the value 1 for this column to distinguish it from subsequent Mass Additions for a conversion, or use the same number for all mass additions from your payables system that you load at once.

CREATED_BY	NUMBER(15)	REQUIRED
-------------------	-------------------	-----------------

Use the value 1 in this column, or the specific user id of the person working on the import.

CREATION_DATE	DATE	REQUIRED
----------------------	-------------	-----------------

Use this column for the date you load this asset into the FA_MASS_ADDITIONS table.

DATE_ PLACED_IN_SERVICE	DATE	REQUIRED PREP
------------------------------------	-------------	----------------------

Use this column for the date you placed the asset in service.

DEPRECIATE_FLAG	VARCHAR2(3)	REQUIRED PREP
------------------------	--------------------	----------------------

Use this column to indicate if you want Oracle Assets to depreciate this asset. If you want to calculate depreciation on this asset, enter YES in this column. If you do not want to calculate depreciation, enter NO.

DEPRN_RESERVE	NUMBER	OPTIONAL
----------------------	---------------	-----------------

Use this column for the depreciation reserve of the asset as it currently appears in your general ledger. You can enter a value, or have Oracle Assets calculate it.

DESCRIPTION	VARCHAR2(80)	REQUIRED PREP
--------------------	---------------------	----------------------

Use this column for a description of the asset.

EXPENSE_CODE_ COMBINATION_ID	NUMBER(15)	REQUIRED PREP
---	-------------------	----------------------

Use this column for the code combination id of the general ledger account to which depreciation expense should be charged. See Step 3 for more information.

FEEDER_ SYSTEM_NAME	VARCHAR2(40)	OPTIONAL
--------------------------------	---------------------	-----------------

Use this column to note that this asset came from a import process. Use a word like CONVERSION or PAYABLES to denote the nature of the import.

FIXED_ASSETS_COST	NUMBER	REQUIRED PREP
--------------------------	---------------	----------------------

Use this column for the cost of the asset as it currently appears in your general ledger.

FIXED_ASSETS_UNITS	NUMBER(15)	REQUIRED PREP
---------------------------	-------------------	----------------------

Use this column for the number of units that make up the asset. You must use a positive integer value.

FULLY_RSVD_		
REVALS_COUNTER	NUMBER	OPTIONAL

Use this column for the number of times you have revalued this asset as fully reserved. If you do not revalue your assets, do not use this column.

INVOICE_CREATED_BY	NUMBER(15)	NULL
---------------------------	-------------------	-------------

Do not use this column.

INVOICE_DATE	DATE	NULL
---------------------	-------------	-------------

Do not use this column.

INVOICE_ID	NUMBER(15)	OPTIONAL
-------------------	-------------------	-----------------

Use this column for the invoice id, if available. Use the appropriate value from the INVOICE_ID column of the AP_INVOICES table. Note that INVOICE_ID is the unique internal identifier, and not the same as the external identifier INVOICE_NUMBER.

INVOICE_NUMBER	VARCHAR2(15)	OPTIONAL
-----------------------	---------------------	-----------------

Use this column for the invoice number for this asset, if available.

INVOICE_UPDATED_BY	NUMBER(15)	NULL
---------------------------	-------------------	-------------

Do not use this column.

LAST_UPDATE_DATE	DATE	REQUIRED
-------------------------	-------------	-----------------

Use this column for the date you load this asset into the FA_MASS_ADDITIONS table.

LAST_UPDATE_LOGIN	NUMBER(15)	REQUIRED
--------------------------	-------------------	-----------------

Use the value 1 in this column, or the specific user id of the person working on the import.

LAST_UPDATED_BY	NUMBER	REQUIRED
------------------------	---------------	-----------------

Use the value 1 in this column, or the specific user id of the person working on the import.

LOCATION_ID	NUMBER(15)	REQUIRED PREP
--------------------	-------------------	----------------------

Use this column for the location id that corresponds with the location of the asset. See Step 5 for more information.

MASS_ADDITION_ID	NUMBER(15)	REQUIRED
-------------------------	-------------------	-----------------

Oracle Assets uses this column as a unique identifier for mass additions. The values in MASS_ADDITION_ID are generally sequential. You can use a sequence generator to populate this column.

MANUFACTURER_ NAME	VARCHAR2(30)	OPTIONAL PREP
-------------------------------------	---------------------	----------------------

Use this column for the name of the manufacturer of the asset.

MERGE_ INVOICE_NUMBER	VARCHAR2(50)	OPTIONAL
--	---------------------	-----------------

Optionally set this column to INVOICE_NUMBER for unmerged lines.

MERGE_ VENDOR_NUMBER	VARCHAR2(30)	OPTIONAL
---------------------------------------	---------------------	-----------------

Optionally set this column to VENDOR_NUMBER for unmerged lines.

MODEL_NUMBER	VARCHAR2(40)	OPTIONAL PREP
---------------------	---------------------	----------------------

Use this column for the model number of the asset.

NEW_MASTER_FLAG	VARCHAR2(3)	NULL
------------------------	--------------------	-------------

Do not use this column.

PARENT_ASSET_ID	NUMBER(15)	OPTIONAL
------------------------	-------------------	-----------------

If you are importing asset information from another payables system, use this column for the asset id of the parent asset if this asset is a subcomponent of another asset. The parent asset must be an existing asset. Note that ASSET_ID is the unique internal identifier, and not the same as the external identifier ASSET_NUMBER.

For a conversion, do not use this column. If an asset is a subcomponent, you can enter this relationship in the Additions form after you post.



Additions
(Oracle Assets Reference Manual)

PARENT_		
MASS_ADDITION_ID	NUMBER(15)	NULL

Do not use this column.

PAYABLES_		
BATCH_NAME	VARCHAR2(50)	NULL

Do not use this column.

PAYABLES_CODE_		
COMBINATION_ID	NUMBER(15)	REQUIRED

Use this column for the code combination id of the asset clearing account you assigned to the asset category. See Step 3 for more information.

PAYABLES_COST	NUMBER	REQUIRED
----------------------	---------------	-----------------

Use this column for the original cost of the asset. If you do not have the original cost of the asset, use the asset cost as it appears in the general ledger.

PAYABLES_UNITS	NUMBER	REQUIRED
-----------------------	---------------	-----------------

Use this column for the number of units that make up the asset. You must use a positive integer value. Use the same value in this column that you use in the FIXED_ASSETS_UNITS column.

PO_NUMBER	VARCHAR2(15)	OPTIONAL PREP
------------------	---------------------	----------------------

Use this column for the purchase order number for this asset, if available.

PO_VENDOR_ID	NUMBER(15)	OPTIONAL
---------------------	-------------------	-----------------

Use this column for the vendor id. See Step 6 for more information.

POST_BATCH_ID	NUMBER(15)	NULL
----------------------	-------------------	-------------

Do not use this column.

POSTING_STATUS	VARCHAR2(15)	REQUIRED
-----------------------	---------------------	-----------------

To import asset information from another payables system, use the value NEW or ON HOLD for this column if you want to review this asset in the Prepare Mass Additions form.

Use the value POST for this column if you are entering all required information and want to post the asset immediately, such as for a conversion.

PRODUCTION_CAPACITY	NUMBER	OPTIONAL
----------------------------	---------------	-----------------

Use this column for the capacity of a units of production asset. If you do not enter a capacity, Oracle Assets uses the capacity from the asset category. If this asset is not a units of production asset, do not use this column.

QUEUE_NAME	VARCHAR2(15)	REQUIRED
-------------------	---------------------	-----------------

Use the same value you entered into the POSTING_STATUS column (e.g. NEW, ON HOLD, POST).

REVAL_ AMORTIZATION_BASIS	NUMBER	OPTIONAL
----------------------------------	---------------	-----------------

Use this column for the basis for amortization of revaluation reserve for a revalued asset, usually the current revaluation reserve. If you do not revalue your assets, do not use this column.

REVAL_RESERVE	NUMBER	OPTIONAL
----------------------	---------------	-----------------

Use this column for the revaluation reserve of a revalued asset. If you do not revalue your assets, do not use this column.

REVIEWER_ COMMENTS	VARCHAR2(60)	OPTIONAL
---------------------------	---------------------	-----------------

Use this column to note that this asset came from a import process and any other details about this asset.

<u>SALVAGE_VALUE</u>	<u>NUMBER</u>	<u>OPTIONAL</u>
-----------------------------	----------------------	------------------------

Use this column for the salvage value of the asset, if available.

<u>SERIAL_NUMBER</u>	<u>VARCHAR2(35)</u>	<u>OPTIONAL PREP</u>
-----------------------------	----------------------------	-----------------------------

Use this column for the serial number of the asset.

<u>SPLIT_MERGED_CODE</u>	<u>VARCHAR2(3)</u>	<u>NULL</u>
---------------------------------	---------------------------	--------------------

Do not use this column.

<u>TAG_NUMBER</u>	<u>VARCHAR2(15)</u>	<u>OPTIONAL PREP</u>
--------------------------	----------------------------	-----------------------------

Use this column for the asset tag number if required. You can use this column for barcode values if you track assets by barcodes. Oracle Assets allows no duplicate values except null.

<u>UNIT_OF_MEASURE</u>	<u>VARCHAR2(25)</u>	<u>OPTIONAL</u>
-------------------------------	----------------------------	------------------------

Use this column for the unit of measure for a units of production asset. If you do not enter a unit of measure, Oracle Assets uses the unit of measure from the asset category. If this asset is not a units of production asset, do not use this column.

<u>UNREVALUED_COST</u>	<u>NUMBER</u>	<u>OPTIONAL</u>
-------------------------------	----------------------	------------------------

Use this column for the cost without regard to any revaluations of a revalued asset. If you do not revalue your assets, do not use this column.

<u>VENDOR_NUMBER</u>	<u>VARCHAR2(15)</u>	<u>NULL</u>
-----------------------------	----------------------------	--------------------

Do not use this column.

<u>YTD_DEPRN</u>	<u>NUMBER</u>	<u>OPTIONAL</u>
-------------------------	----------------------	------------------------

Use this column for the year-to-date depreciation of the asset as it currently appears in your general ledger. You can enter a value, or have Oracle Assets calculate it for an asset with a date placed in service in a prior period.

YTD_REVAL_ DEPRN_EXPENSE	NUMBER	OPTIONAL
-------------------------------------	---------------	-----------------

Use this column for the year-to-date depreciation expense due to revaluation of a revalued asset. If you do not revalue your assets, do not use this column.

Step 2 Load Your Asset Data Into The FA_MASS_ADDITIONS Table

You use SQL*Plus to load your asset information into the FA_MASS_ADDITIONS table from the interim table. Be sure to check the table after each SQL*Plus script to ensure that the script updated the table correctly. Also load the LAST_UPDATE_DATE column and all the other columns that are the same for all your assets.



Suggestion: Load your data into the FA_MASS_ADDITIONS table in stages, posting and cleaning the table, to avoid exceeding tablespace allocations.

Step 3 Load Expense Code Combination IDs

Use SQL*Plus to match expense account information in your interim table with the correct segments of the GL_CODE_COMBINATIONS table. To do this, you must first determine the mapping between segment numbers and segment names. Find the name of your chart of accounts using the Define Set of Books form to query on the set of books that you entered for the depreciation book in the Book Controls form. When you know the chart of accounts, perform the following SQL*Plus script:

Example:

```
select segment_name, application_column_name
  from fnd_id_flex_segments
 where id_flex_code = 'GL#'
 and enabled_flag = 'Y'
 and id_flex_num in (
   select id_flex_num
   from fnd_id_flex_structures
  where id_flex_structure_name =
        Your Chart of Accounts Name
 and id_flex_code = 'GL#')
```

Match the information you have in the interim table with the appropriate segments to determine the correct code combination id for each asset. You might have only an account in your interim table, or you might have a company, division, and cost center that you need to match with segments in the GL_CODE_COMBINATIONS table. Make certain that your SQL*Plus script selects only one code combination id for each asset.

You can create the combinations you need using the Define Accounting Flexfield Combinations form.

Step 4 Load Category IDs

The asset category determines many of the accounts to which each asset belongs. The category and date placed in service also determine the depreciation method, prorate convention, and other depreciation rules for the asset based on default values you defined for each category. Unless the interim table contains explicit information about the category to which each asset belongs, you must use all the information you have about each asset to determine its category. Asset account and reserve account are often useful in determining an asset's major category.

You need to determine the name of the category flexfield structure using the System Controls form. When you know the name of the category flexfield, perform the following SQL*Plus script:

Example:

```
select segment_name, application_column_name
  from fnd_id_flex_segments
 where id_flex_code = 'CAT#'
 and enabled_flag = 'Y'
 and id_flex_num in (
   select id_flex_num
   from fnd_id_flex_structures
  where id_flex_structure_name =
         Your Category Flexfield Name
 and id_flex_code = 'CAT#')
```

Match the information you have with the corresponding columns in the FA_CATEGORIES table, the FA_CATEGORY_BOOKS table, and the FA_CATEGORY_BOOK_DEFAULTS table. These tables join using the CATEGORY_ID column in each. The FA_CATEGORY_BOOKS table contains information about a category that is specific to a book, e.g. accounts. The FA_CATEGORY_BOOK_DEFAULTS table contains information about a category that is specific to a book and date placed in service range, e.g. depreciation method. The FA_CATEGORIES table contains information about a category that is common for all books, including the category flexfield segment values. You can match on segments the same way you do for the Expense Code Combination ID if you have enough information in the interim table.

You can set up categories using the Asset Categories form.

Step 5 **Load Location IDs**

Use SQL*Plus to match location information in your interim table with the location segments in the FA_LOCATIONS table. To do this, you need to determine the mapping between segment names and segment numbers. Determine the name of the location flexfield structure you defined on the System Controls form and then perform the following SQL*Plus script:

Example:

```
select segment_name, application_column_name
  from fnd_id_flex_segments
 where id_flex_code = 'LOC#'
 and enabled_flag = 'Y'
 and id_flex_num in (
   select id_flex_num
   from fnd_id_flex_structures
  where id_flex_structure_name =
         Your Location Flexfield Name
 and id_flex_code = 'LOC#')
```

Match the location information in your interim table with the location segments in the FA_LOCATIONS table. Load the LOCATION_ID of the matching location record into the FA_MASS_ADDITIONS table. Be certain you select only one location id for each asset.

You can set up locations using the Locations form.

Step 6 **Load Vendor Information**

If you have vendor information for your assets, use SQL*Plus to match the vendor name in the interim table with the PO_VENDOR_NAME column in the PO_VENDORS table. Load the PO_VENDOR_ID of the matching record into the FA_MASS_ADDITIONS table.

You need to set up vendors in the Enter Vendor form if you are not using Oracle Payables or Oracle Purchasing. Note that PO_VENDOR_ID is the unique internal identifier, and not the same as the external identifier vendor number.

Step 7 **Run Mass Additions Prepare, If Necessary, And Mass Additions Post**

Once the FA_MASS_ADDITIONS table is loaded, you must change the POSTING_STATUS column to HOLD for any assets that have a date placed in service after the end of the conversion period. For most

conversions, the conversion period is the last period of the previous fiscal year.

When you are ready to post the FA_MASS_ADDITIONS table, use the Send Mass Additions to Oracle Asset form to run Mass Additions Post. This program moves your assets into Oracle Assets.



Send Mass Additions to Oracle Assets
(*Oracle Assets Reference Manual*)

Run the Mass Additions Status Report to see the results. Compare this report with the expected results, and investigate missing or incomplete items. Check the PERIOD_FULLY_RESERVED column in FA_BOOKS for fully reserved assets.

For assets that you placed in service after the conversion period, you can run Mass Additions Post after you have opened the appropriate period. For example, if your fiscal year ends December 31, December is your conversion period. If you have some assets that you placed in service in January, set the posting status to ON HOLD for the January assets while you post the first time. After you run depreciation for December, then you set the posting status of the January assets to POST and run Mass Additions Post again.

Step 8 Fix Exceptions

During this step you fix all the exceptions which were not properly imported using Mass Additions. You only need to perform these steps if they apply to your import:

1. Assets With Multiple Distributions:

If you want some of your assets to have multiple distribution lines, then you need to use the Transfers form to correct the distribution information for each of these assets.



Transfers
(*Oracle Assets Reference Manual*)

2. Assets With Investment Tax Credits:

After you have posted your assets with Mass Additions, use the Assign Investment Tax Credit form to add ITC information.



Assign Investment Tax Credit
(*Oracle Assets Reference Manual*)

3. Leased Assets And Leasehold Improvements:

After you have posted your assets with Mass Additions Post, use the Additions form to add leasing information for your leased assets and your leasehold improvements. Verify that the life of your leasehold improvements is correct when you run depreciation.

4. Assets With Parent Or Child Assets

After you have posted your assets using Mass Additions Post, use the Additions form to add parent asset information to each child asset.



Additions

(Oracle Assets Reference Manual)

Finishing Your Import

When you have all your asset information moved into Oracle Assets, you need to verify that the financial information matches your records. Once you are satisfied that the value of your assets is correct, you can run depreciation and then verify that the accumulated depreciation matches your records. When you are satisfied that your corporate book is correct, you can copy the assets into your tax books. You reconcile each tax book using a similar procedure.

If some of the assets in the FA_MASS_ADDITIONS table have dates placed in service after the import, you need to bring these assets into Oracle Assets in the correct period.

Finishing Checklist

Complete the following steps to finish your import:

- ☐ 1. Verify your assets
- ☐ 2. Run Depreciation
- ☐ 3. Reconcile depreciation amounts
- ☐ 4. Run Mass Additions for post-dated assets, if necessary
- ☐ 5. Copy assets to your tax books using Mass Copy
- ☐ 6. Reconcile your tax books
- ☐ 7. Clean up the Mass Additions holding area

Finishing Steps

Step 1 Verify Your Assets

Before you run depreciation you should run the Asset Additions Report. Use this report to verify that each asset has the correct depreciation method, life, and date placed in service. Also verify that each asset has the correct cost and depreciation reserve and that the totals for each asset account are correct. If you find any errors, make

adjustments using the Depreciation Books form and reclassifications using the Additions form.



Asset Additions Report
(*Oracle Assets Reference Manual*)

For additional verification, project depreciation to the asset and cost center level to see that the expense projections agree with your estimates, and that the assets appear properly.



Depreciation Projections
(*Oracle Assets Reference Manual*)

Step 2 Run Depreciation

When you are satisfied that your assets are correct, run depreciation for the conversion period. After depreciation completes, Oracle Assets automatically runs the Journal Entry Reserve Ledger report.



Run Depreciation
(*Oracle Assets Reference Manual*)

Step 3 Reconcile Depreciation Amounts

Use the Journal Entry Reserve Ledger Report from Step 2 to verify that the depreciation amounts are correct. If Oracle Assets calculated depreciation for you, verify that the calculated amount is correct. If you find any errors, make adjustments using the Depreciation Books form and reclassifications using the Additions form.

Step 4 Run Mass Additions For Post-Dated Assets, If Necessary

If necessary, run Mass Additions to add assets into the periods following your import period. Add any other new assets and perform any transactions that you made during the period. Verify all these transactions and run depreciation again. Repeat this procedure until you have caught up to the current period.

Step 5 Copy Your Assets To Your Tax Books Using Mass Copy

When you are satisfied that your corporate book is correct, use Mass Copy to copy your assets into your tax books. You should set up your tax books so that the first period starts at the same time as the associated corporate book. If your import period is the last period of the previous fiscal year, use Initial Mass Copy. If your import is the

first period of the current fiscal year, use Periodic Mass Copy since there is no historical data in Oracle Assets.



Using Mass Copy
(*Oracle Assets Reference Manual*)

Step 6 Reconcile Your Tax Books

Reconcile your tax books the same way you did your corporate book, but use the Tax Reserve Ledger Report in place of the Journal Entry Reserve Ledger Report.

Run Periodic Mass Copy each period to bring over any new assets, cost adjustments, retirements, and reinstatements from the corporate book.

Step 7 Clean Up The Mass Additions Holding Area

After you have successfully imported a group of assets, you should remove them from the mass additions holding area. First, run the Unposted Mass Additions Report and verify the status of any unposted mass additions. Afterward, use the Delete Mass Additions form, and the Purge Mass Additions form if necessary.



Delete Mass Additions
Purge Mass Additions
(*Oracle Assets Reference Manual*)

Integrating Oracle Assets Using the Budget Interface

Oracle Assets allows you to enter budget information manually, or you can maintain your budget information in another system and upload the information using the budget interface. You prepare and analyze your budget information on any feeder system and then automatically transfer it into Oracle Assets. You can use this information to project depreciation expense for your capital budgets and compare actual and planned capital spending in Oracle Assets.



Overview of Capital Budgeting
(*Oracle Assets Reference Manual*)

Basic Business Needs

Oracle Assets lets you:

- Maintain budget information in the environment you prefer
- Transfer budget information from any feeder system to your Oracle Assets database using the Upload Capital Budget form
- Review budget amounts
- Update budget information online if necessary

Terms

If you are not already familiar with the following terms, you may wish to consult the Glossary in your Oracle Assets Reference Manual to learn more about them.

- Budget Book
- Budget Interface Table
- Budget Upload
- Budget Worksheet

Major Features

Spreadsheet Compatibility

If you maintain your budget information in a spreadsheet, you can upload it to Oracle Assets using the budget interface. You can transfer budget data from any software package that prints to an ASCII file, and then use SQL*Loader to load the FA_BUDGET_INTERFACE table described later in this essay. The budget interface is compatible with spreadsheets such as SQL*Calc and Lotus 1-2-3 with the Oracle Add-in for 1-2-3.

Update Budget Amounts

If you need to update any budget amounts, use the Enter Capital Budget form. You can also delete the existing budget information in any budget book and reload the information.

Prerequisites

Before using Capital Budgeting, you must set up the depreciation books you want to use for Capital Budgeting.



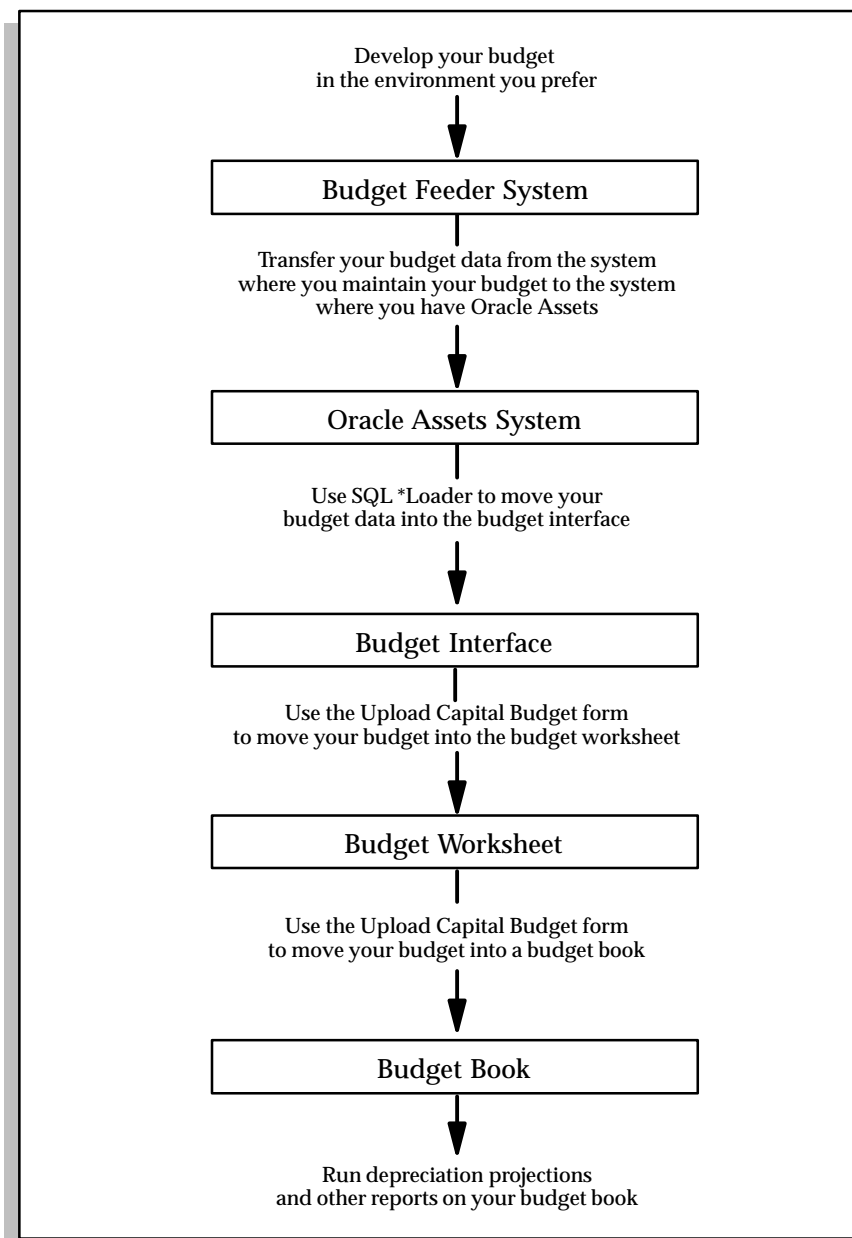
Book Controls

(Oracle Assets Reference Manual)



Suggestion: Your budget book must use the same calendar as the corporate book associated with it so you can compare actual and budgeted spending.

**Figure 2 – 1 Flow of the
Capital Budgeting Process**



Entering Budget Information

You can either enter capital budget information directly into Oracle Assets or upload budget information you created on another system.

Online Budget Entry

Use the Enter Capital Budget form to enter or modify your budgets online. Enter the budget amount for planned acquisitions for each cost center and asset category. You can then use the Upload Capital Budget form to load your budget into a budget book so you can run reports and project depreciation expense on your capital budget.



Upload Capital Budget
Enter Capital Budget
(*Oracle Assets Reference Manual*)

Uploading Budgets From Other Systems

Uploading budgets from other systems (such as a spreadsheet on a personal computer) into Oracle Assets is a five step process.

Step 1: Use a file transfer program such as Kermit to upload your ASCII budget file from your personal computer to the computer where you have Oracle Assets.

Step 2: Use SQL*Loader to move your budget into the Budget Interface. There is a sample SQL*Loader script in this essay which you need to modify.

Step 3: Use the Upload Capital Budget form to move your budget into the Budget Worksheet. Enter Yes in the Delete Existing Budget field if you are replacing an existing budget.

Step 4: Use the Enter Capital Budget form to review or change your budget.

Step 5: Use the Upload Capital Budget form to move your budget into a budget book.

Upload Your Spreadsheet Directly Into The Budget Interface

If you work in Lotus 1–2–3 with the Oracle Add-in for 1–2–3, you can insert the rows of your spreadsheet into the budget interface table by simply using menu commands. Oracle Add-in for 1–2–3 automates Step 2.



Additional Information: *Oracle Add-in for 1–2–3 Manual*

Budget Interface Table

FA_BUDGET_INTERFACE, the budget interface table, is organized into columns in which Oracle Assets stores budget information. It contains the following columns:

<u>Column Name</u>	<u>Type – Description</u>
BOOK_TYPE_CODE	Alphanumeric The name of the budget book
PERIOD1_AMOUNT through PERIOD12_AMOUNT	Numeric The budget amount you allocate to an asset category and cost center for one period in your fiscal year. For a budget book, you allocate the annual budget over up to 12 periods.
ACCT_SEGMENT1 through ACCT_SEGMENT30	Numeric Segment of your accounting flexfield.
CAT_SEGMENT1 through CAT_SEGMENT7	Alphanumeric Segment of your category flexfield.

Customizing the SQL*Loader script

Figure 2 – 2 is a sample SQL*Loader script (filename: *budget.ctl*) and Figure 2 – 3 is a budget information data file (filename: *budget.dat*) that Oracle Assets uses for capital budgeting. You can easily modify this script to load your budget into the Budget Interface. This script only expects the company, cost center, and major category. Modify it to accept whatever detail you provide.

Figure 2 – 2 Sample SQL*Loader script

```
LOAD DATA
INFILE budget.dat
INTO TABLE FA_BUDGET_INTERFACE
FIELDS TERMINATED BY WHITESPACE

(BOOK_TYPE_CODE CONSTANT 'FY90',

PERIOD1_AMOUNT,
PERIOD2_AMOUNT,
PERIOD3_AMOUNT,
PERIOD4_AMOUNT,
PERIOD5_AMOUNT,
PERIOD5_AMOUNT,
PERIOD6_AMOUNT,
PERIOD7_AMOUNT,
PERIOD8_AMOUNT,
PERIOD9_AMOUNT,
PERIOD10_AMOUNT,
PERIOD11_AMOUNT,
PERIOD12_AMOUNT,

ACCT_SEGMENT1,
ACCT_SEGMENT2,
ACCT_SEGMENT3 CONSTANT '0000',
ACCT_SEGMENT4 CONSTANT '000',
ACCT_SEGMENT5 CONSTANT '000',
ACCT_SEGMENT6 CONSTANT '0000',

CAT_SEGMENT1,
CAT_SEGMENT2 CONSTANT 'NONE')
```

Company Code (nn)	Accounting Flexfield Structure
Cost Center (nnn)	
Account (nnnn)	
Product (nnn)	
Product Line (nnn)	
Sub-Account (nnnn)	
Primary Category	Category Flexfield Structure
Subcategory	

Figure 2 – 3 Sample budget data file

```
12000 15000 15000 15000 36000 20000 20000 20000 20000 20000 15000 15000 01 100 COMPUTER
15000 12000 17500 25000 28000 28000 28000 28000 15000 15000 15000 20000 01 200 AUTO
```

The SQL*Loader script reads in three kinds of data: budget amounts, general ledger numbers, and categories.

Budget Amounts

You must define a PERIOD#_AMOUNT in the SQL*Loader script for each period of your corporate calendar.

Therefore, edit the SQL*Loader script to:

- Reflect the number of periods in your company's corporate calendar

General Ledger Number

You must define an ACCT_SEGMENT# in the SQL*Loader script for each segment in your accounting flexfield. In this case, only the company and cost center are specified in the data file. The other ACCT_SEGMENT#s are held constant in the control file with zeros as place holders.

Therefore, edit the SQL*Loader script to:

- Reflect the number of segments in your company's accounting flexfield
- Reflect the position of each segment in your company's accounting flexfield

When creating your budget data file, make sure that each segment value has the correct number of digits. For example, if your cost center consists of three digits, you must enter cost centers 5 and 25 as 005 and 025.

Categories

You must define a CAT_SEGMENT# in the SQL*Loader script for each segment in your category flexfield. In this example, only the major category is specified in the data file. The other CAT_SEGMENT#s are held constant in the control file with the word NONE. Note that each asset category must be defined with subcategory NONE for the corporate book in Oracle Assets.

Therefore, edit the SQL*Loader script to:

- Reflect the number of segments used in your company's category flexfield
- Reflect the position of each segment used in your company's category flexfield

For your SQL*Loader script to work properly, you must define your segments as one word. For instance, you could define the asset category name Leasehold Improvements as LImprove or Lease_Improvement. For information on how to change the SQL*Loader script to accept asset category names of more than one word, consult the SQL*Loader manual.

You must also change the name of the data file, budget book, and company number in the SQL*Loader script (see italicized words in script). Remember that you must enter names exactly as they appear in Oracle Assets. Thus, do not enter the budget book name fy90 in the script file when your budget book name is actually FY90.

To execute the SQL*Loader script and load your budget data into the Budget Interface, type the following at the system prompt:

```
sqlload <account_name/password> control=budget.ctl
```

Integrating Oracle Assets Using the Production Interface

Oracle Assets allows you to enter production information manually, or you can maintain your production information in another system and upload the information using the production interface. Prepare and analyze your production information on any feeder system and then automatically transfer your production information into Oracle Assets. Oracle Assets uses that information to calculate depreciation for your units of production assets.



Overview of Units Of Production
(*Oracle Assets Reference Manual*)

Basic Business Needs

Oracle Assets allows you to:

- Maintain production information in the environment you prefer
- Transfer production information from any feeder system to your Oracle Assets database using the Production Upload form
- Review production amounts
- Update production online if necessary

Terms

If you are not already familiar with the following terms, you may wish to consult the Glossary in your Oracle Assets Reference Manual to learn more about them.

- Production Upload
- Production Interface Table

Major Features

Spreadsheet Compatibility

If you maintain your production information in a spreadsheet, you can upload it to Oracle Assets using the production interface. You can transfer production data from any software package that prints to an ASCII file, and then use SQL*Loader to load the FA_PRODUCTION_INTERFACE table described later in this essay. The production interface is compatible with spreadsheets such as SQL*Calc and Lotus 1-2-3 with the Oracle Add-in for 1-2-3.

Production History Report

After you have uploaded production, you can run the Production History Report. Use this report to review production amounts for your units of production assets.

Update Production

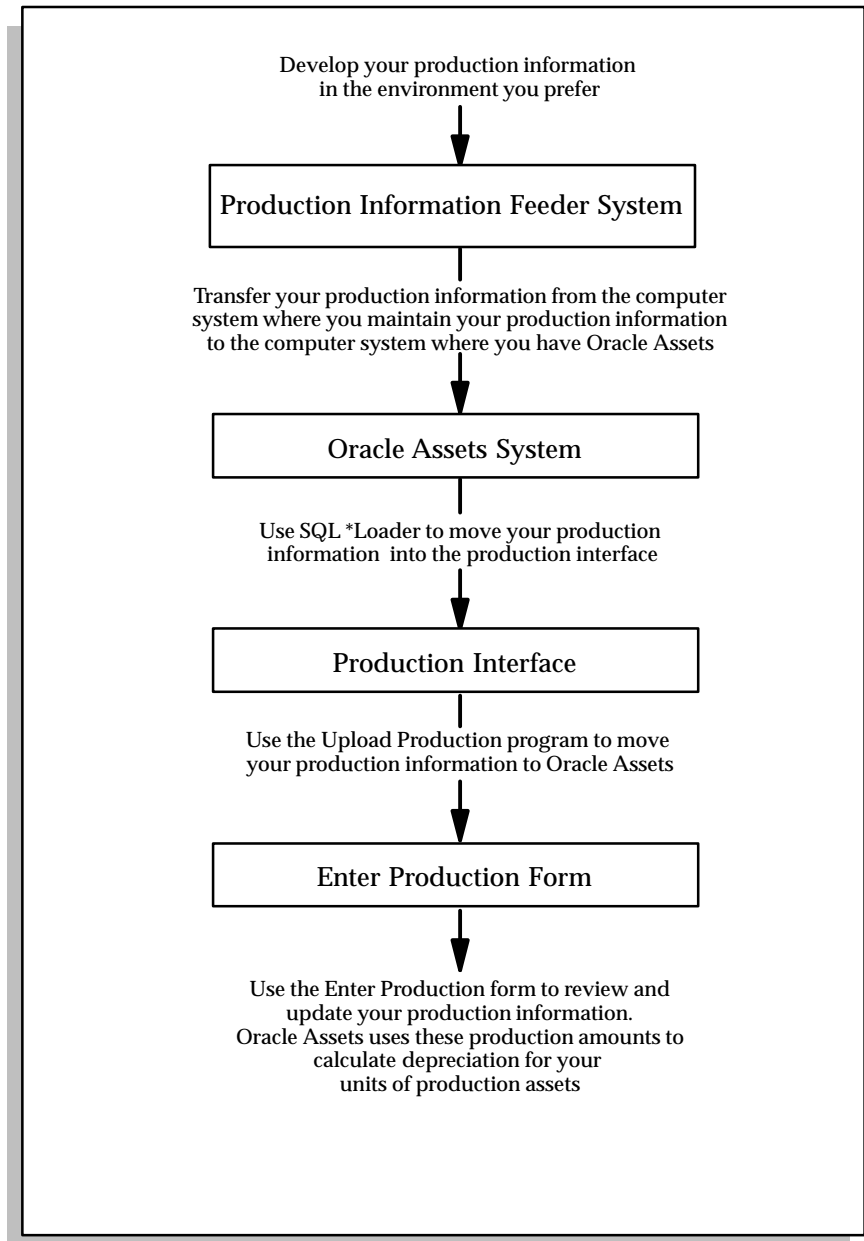
Use the Enter Production form to update any production amounts. If you update the production information in your spreadsheet, you can upload production again for any periods for which you have not yet run depreciation. If you upload production again, you must enter production amounts for the same date ranges you entered before. Use the Enter Production form if you need to change both the production amount and the date range.

Production Validation

The production interface validates your production information for compatibility with Oracle Assets. The production interface checks to make sure that the following conditions are true:

- Your asset is a capitalized units of production asset
- The date range falls within a depreciation period
- You haven't already entered production for this date
- You aren't entering production for overlapping date ranges
- If you enter a negative production amount, the life-to-date production for the asset cannot be less than zero
- The date range cannot be for a closed period

Figure 2 – 4 Flow of the Upload Production Process



Entering Production Information

You can either enter production information directly into Oracle Assets or upload production information you entered on another system.

Online Production Entry

Use the Enter Production form to enter or modify your production amounts online. Enter the production amount for an asset for specified dates. Oracle Assets uses these amounts to calculate depreciation for your units of production assets.



Enter Production
(*Oracle Assets Reference Manual*)

Uploading Production From Other Systems

Uploading production amounts from other systems (such as a spreadsheet on a personal computer) into Oracle Assets is a four step process.

Step 1: Use a file transfer program such as Kermit to upload your ASCII production information file from your personal computer to the computer where you have Oracle Assets, if necessary.

Step 2: Use SQL*Loader to move your production amounts into the production interface table. There is a sample SQL*Loader script in this essay.

Step 3: Use the Upload Production form to move your production information into Oracle Assets.



Upload Asset Production
(*Oracle Assets Reference Manual*)

Step 4: Use the Enter Production form to review or change your production information.

Upload Your Spreadsheet Directly Into The Production Interface

If you work in Lotus 1-2-3 with the Oracle Add-in for 1-2-3, you can insert the rows of your spreadsheet into the production interface table by simply using menu commands. Oracle Add-in for 1-2-3 automates Step 2.



Additional Information: *Oracle Add-in for 1-2-3 Manual*

Production Interface Table

FA_PRODUCTION_INTERFACE, the production interface table, is organized into columns in which Oracle Assets stores production information. It contains the following required columns:

<u>Column Name</u>	<u>Type – Description</u>
ASSET_NUMBER	Alphanumeric The asset number of the units of production asset for which you want to enter production.
PRODUCTION	Numeric The production amount for the asset between Start_Date and End_Date.
START_DATE	Date The first date to which this production amount applies.
END_DATE	Date The last date to which this production amount applies.

The start date and end date must fall within a single depreciation period in Oracle Assets.

You cannot enter production for dates before the asset's prorate date in the corporate book or for a period for which you have already run depreciation.

You also cannot enter production for date ranges which overlap. For example, if you enter production for 01-JUN-1993 through 15-JUN-1993, you cannot enter production for 15-JUN-1993 through 30-JUN-1993. You can enter production for 16-JUN-1993 through 30-JUN-1993.

You cannot enter production for periods prior to the current open period in your corporate book.

Customizing the SQL*Loader script

Listed below is a sample SQL*Loader script (filename: *production_information.ctl*) and production information data file (filename: *production_information.dat*) that Oracle Assets uses for units of production depreciation. You can easily modify this script to load your production information into the production interface.

You must enter the start and end dates in your data file in the same format which you specify in the SQL*Loader script.

Figure 2 – 5 Sample SQL*Loader script

```
LOAD DATA
INFILE production_information.dat
INTO TABLE FA_PRODUCTION_INTERFACE
FIELDS TERMINATED BY WHITESPACE

(ASSET_NUMBER,
PRODUCTION,
START_DATE DATE "DD-MON-YYYY",
END_DATE DATE "DD-MON-YYYY")
```

Figure 2 – 6 Sample Production Information data file

321456	1000	01-JUL-1993	31-JUL-1993
421345	1100	01-AUG-1993	05-AUG-1993
322534	1200	16-AUG-1993	26-AUG-1993
323242	1300	24-AUG-1993	31-AUG-1993
234261	1400	01-SEP-1993	30-SEP-1993
433251	1500	01-OCT-1993	13-NOV-1993

Production Information

For each asset and date range for which you want to enter a production amount, you must specify the following information in the SQL*Loader script:

- Asset Number
- Production Amount
- Start Date
- End Date



Attention: You must enter the start and end dates in your data file in the same format which you specify in the SQL*Loader script.

Remember that you must enter the asset number exactly as it appears in Oracle Assets. Thus, be careful not to enter the asset number uopasset1 in the script file when your asset number is actually UOPASSET1.

Running SQL*Loader

To execute the SQL*Loader script and load your production data into the Production Interface, type the following at the system prompt:

```
$ sqlload <account_name/password>  
control = production_information.ctl
```

Integrating Oracle Assets Using the ACE Interface

You can either have Oracle Assets calculate ACE information for you, or enter it yourself. If you want Oracle Assets to calculate ACE depreciation reserve for you, enter historical asset information beginning no later than the last period of fiscal 1989. Start your depreciation books in Oracle Assets before the end of fiscal 1989, and enter asset transactions through the current period.

If you have ACE information from another asset system, you can load it into Oracle Assets using the ACE interface. Define the initial open period of your ACE book as the last period of the last fiscal year you completed on your previous system. Then load the depreciation reserve for your ACE assets using this interface.

Basic Business Needs

Oracle Assets allows you to:

- Create an ACE book
- Provide historical asset information for Oracle Assets to calculate ACE information for you
- Alternatively, load existing ACE information into Oracle Assets from another system
- Update ACE book with ACE information

Terms

If you are not already familiar with the following terms, you may wish to consult the Glossary in your Oracle Assets Reference Manual to learn more about them.

- ACE Book

ACE Update Process

Create An ACE Book

Create an ACE tax book with ACE depreciation rules. Copy your assets into the ACE book from the corporate book.



ACE Coverage in Oracle Assets
(*Oracle Assets Reference Manual*)

Populate The ACE Conversion Table

Either use the Define ACE Book form to have Oracle Assets populate the ACE conversion table, or load the table manually with the ACE information from your previous system. The rest of this essay discusses the information you must load into the conversion table.

Update The ACE Book

Use the Define ACE Book form to update the assets in the ACE book with the information in the ACE conversion table.



Define ACE Book
(*Oracle Assets Reference Manual*)

ACE Conversion Table

FA_ACE_BOOKS, the ACE conversion table, is organized into columns in which Oracle Assets stores ACE information. It contains the following columns:

<u>Column Name</u>	<u>Type – Description</u>
ASSET_ID	Numeric – Asset identification number
MLC_UPDATE_FLAG	Alphanumeric – YES if update ACE book program needs to update the depreciation method, life, prorate convention, and rate adjustment factor with the values in this table
COST	Numeric – Current asset cost

ADJUSTED_ COST	Numeric – Depreciable basis of the asset, calculated as recoverable cost less the depreciation reserve
DEPRN_ METHOD_CODE	Alphanumeric – Depreciation method for the asset
LIFE_ IN_MONTHS	Numeric – Whole life of the asset in months
PRORATE_ CONVENTION_ CODE	Alphanumeric – Prorate convention for the asset
RATE_ ADJUSTMENT_ FACTOR	Numeric – Internal factor used to adjust depreciation rates to depreciate the net book value over the remaining life
YTD_DEPRN	Numeric – Not currently used
DEPRN_ RESERVE	Numeric – Starting depreciation reserve for the asset
PRODUCTION_ CAPACITY	Numeric – Production capacity for assets that use a units of production method
ADJUSTED_ CAPACITY	Numeric – Remaining capacity to use for depreciation for assets that use a units of production method
LTD_ PRODUCTION	Numeric – Life-to-date production for assets that use a units of production method
ADJUSTED_ RATE	Numeric – Not currently used
BASIC_RATE	Numeric – Not currently used

Using Oracle Assets to Load the ACE Conversion Table

When you use Oracle Assets to populate the ACE conversion table, it uses the existing asset information in your federal and AMT tax books to load the table. Cost, depreciation rule, and rate adjustment factor information comes from the FA_BOOKS row that was active for the source tax book at the beginning of fiscal 1990. Depreciation reserve information comes from the FA_DEPRN_SUMMARY row for the source tax book for the last period in fiscal 1989.

Oracle Assets creates a row in the table for each asset in the ACE book that you placed in service before the end of fiscal 1989.

For All Assets Placed In Service Before Fiscal 1981 Or Assets With Depreciation Methods Other Than ACRS And MACRS:		
Column	Value	Source Tax Book
ASSET_ID	Asset Identification Number	Federal Book
MLC_UPDATE_FLAG	YES	
COST	Current Cost	Federal Book
ADJUSTED_COST	Adjusted Cost	Federal Book
DEPRN_METHOD_CODE	Depreciation Method	Federal Book
LIFE_IN_MONTHS	Life	Federal Book
PRORATE_CONVENTION_CODE	Prorate Convention	Federal Book
RATE_ADJUSTMENT_FACTOR	Rate Adjustment Factor	Federal Book
DEPRN_RESERVE	Depreciation Reserve	Federal Book at end of fiscal 1989

Table 2 – 1 (Page 1 of 1)

For All Assets Depreciating Under ACRS:		
Column	Value	Source Tax Book
ASSET_ID	Asset Identification Number	Federal Book
MLC_UPDATE_FLAG	NO	
COST	Current Cost	Federal Book
ADJUSTED_COST	Recoverable Cost – Depreciation Reserve	Federal Book
DEPRN_METHOD_CODE	NULL	
LIFE_IN_MONTHS	NULL	
PRORATE_CONVENTION_CODE	NULL	
RATE_ADJUSTMENT_FACTOR	Remaining Life in Months / Whole Life in Months	Federal Book
DEPRN_RESERVE	Depreciation Reserve	Federal Book at end of fiscal 1989

Table 2 – 2 (Page 1 of 1)

**For All Assets Depreciating Under MACRS
Placed In Service Before Fiscal 1990:**

Column	Value	Source Tax Book
ASSET_ID	Asset Identification Number	AMT Book
MLC_UPDATE_FLAG	NO	
COST	Current Cost	AMT Book
ADJUSTED_COST	Recoverable Cost – Depreciation Reserve	AMT Book
DEPRN_METHOD_CODE	NULL	
LIFE_IN_MONTHS	NULL	
PRORATE_CONVENTION_CODE	NULL	
RATE_ADJUSTMENT_FACTOR	Remaining Life in Months / Whole Life in Months	AMT Book
DEPRN_RESERVE	Depreciation Reserve	AMT Book at end of fiscal 1989

Table 2 – 3 (Page 1 of 1)

Manually Loading the ACE Conversion Table

When you manually load the ACE conversion table, you must enter the existing ACE information. Load information for all assets that you placed in service before fiscal 1990.

For All Assets Placed In Service Before Fiscal 1981 Or Assets With Unrecognized Depreciation Methods:		
Column	Value	Source Tax Book
ASSET_ID	Asset Identification Number	Federal Book
MLC_UPDATE_FLAG	YES	
COST	Current Cost	Federal Book
ADJUSTED_COST	Adjusted Cost	Federal Book
DEPRN_METHOD_CODE	Depreciation Method	Federal Book
LIFE_IN_MONTHS	Life	Federal Book
PRORATE_CONVENTION_CODE	Prorate Convention	Federal Book
RATE_ADJUSTMENT_FACTOR	Rate Adjustment Factor	Federal Book
DEPRN_RESERVE	Depreciation Reserve	Federal Book at end of conversion period

Table 2 – 4 (Page 1 of 1)

For all assets placed in service before fiscal 1981 and assets that are not using ACRS or MACRS methods, load asset information from the federal book as of the end of the conversion period. The conversion period is the first period of your ACE book in Oracle Assets.

If you have not performed any amortized adjustments or revaluations on the asset, the adjusted cost is the same as the current cost, and the rate adjustment factor is 1. Otherwise, the adjusted cost is the recoverable cost less the depreciation reserve and the rate adjustment factor is the remaining life divided by the whole life at the time of the adjustment.

**For All Assets Depreciating Under ACRS Or MACRS
Placed In Service Before Fiscal 1994:**

Column	Value	Source Tax Book
ASSET_ID	Asset Identification Number	Federal Book
MLC_UPDATE_FLAG	NO	
COST	Current Cost	Federal Book
ADJUSTED_COST	Recoverable Cost – Depreciation Reserve	Calculated
DEPRN_METHOD_CODE	NULL	
LIFE_IN_MONTHS	NULL	
PRORATE_CONVENTION_CODE	NULL	
RATE_ADJUSTMENT_FACTOR	Remaining Life in Months / Whole Life in Months	Calculated
DEPRN_RESERVE	Depreciation Reserve	Entered

Table 2 – 5 (Page 1 of 1)

For all assets depreciating under ACRS or MACRS that you placed in service before fiscal 1994, load depreciation reserve and cost information from your old ACE system as of the end of the conversion period. The conversion period is the first period of your ACE book in Oracle Assets.

Oracle Cash Management Open Interfaces

The *Oracle Cash Management User's Guide Release 10SC* contains information about the following Oracle Cash Management open interfaces:

- Bank Statement Open Interface
- Reconciliation Open Interface

Oracle General Ledger Applications Open Interfaces

This chapter contains information about the following Oracle General Ledger Applications open interfaces:

- Integrating Oracle General Ledger or Oracle Government General Ledger Using Journal Import
- Integrating Oracle General Ledger or Oracle Government General Ledger Using Budget Upload

Integrating Oracle General Ledger or Oracle Government General Ledger Using Journal Import

Journal Import lets you quickly integrate new or existing applications such as payroll, accounts receivable, accounts payable and fixed assets with your Oracle General Ledger application. For each accounting period, you can import accounting data from these feeder systems, then review, update and post the journal entries. You can also use Journal Import to import historical data from your previous accounting system.

Basic Integration Needs

Journal Import provides you with the features you need to satisfy your basic integration needs. You should be able to:

- Import accounting data quickly, and automatically convert the valid data you import into journal entries which are meaningful to your organization.
- Import accounting data from a variety of environments, including your own environment and other accounting systems. Import historical data from your previous accounting systems to keep your records consistent and up-to-date.
- Run several copies of Journal Import, for the same source or for different sources, at the same time.
- Import journal references to maintain a mapping of how detail Accounting Flexfields are summarized into a single Accounting Flexfield when you choose to create summary journals.
- Import descriptive flexfield structures with your journal information.
- Import government transaction codes associated with your journal lines, if you are using Oracle Government General Ledger.
- Import sequence numbers for detail transactions from your feeder systems. Choose to automatically assign sequential numbers to journal entries created by Journal Import to comply with audit requirements of your country.
- Create journal entries with suspense journal lines when Journal Import detects Accounting Flexfield errors in your source data.

- Display results of your import run. You can identify which data has been successfully imported, and any errors which may have occurred during the import process.
- Correct invalid data online that Journal Import was unable to use to create journal entries.
- Choose to archive your source data each time you run Journal Import.

Major Features

Simple Integration

With Journal Import, you can import accounting data from a variety of sources. You can import accounting data from any Oracle Financials or Oracle Government Financials product, such as Oracle Payables and Oracle Government Payables. You can also import data from any non-Oracle system which maintains your accounting data.

Historical Data Conversion

With Journal Import, you can import accounting data from your previous accounting system, providing a smooth and efficient transition from your old accounting application to your Oracle General Ledger application.

Parallel Processing

Your Oracle General Ledger application allows you to run several copies of Journal Import, for the same source or for different sources, at the same time. This allows you to save time, and further improve throughput on your parallel processing computer.

Complete Validation

Journal Import validates all of your accounting data during the import process. Journal Import also ensures that the imported accounting data is accurate and valid within your Oracle General Ledger application. If you choose to import your descriptive flexfields along with your journal lines, you can specify whether you want Journal Import to validate your descriptive flexfields.

Journal Import Execution Report

Each time you run Journal Import, your Oracle General Ledger application prints a Journal Import Execution Report that tells you the total number of journal entry lines Journal Import successfully imported. The Journal Import Execution Report also informs you of any errors that occurred during the import process. Use the Journal Import Execution Report to identify errors. You can then correct the errors and successfully re-import your data.

Journal Entry Creation

Journal Import creates journal entries from the accounting data you import. You can specify a name for your import journal entry or let Journal Import create a name for you. And, Journal Import ensures that your accounting data is valid before it creates import journal entries.

Journal Reference Conversion

With Journal Import, you can import journal reference information to create a one-to-one mapping of how detail Accounting Flexfields are summarized into a single Accounting Flexfield. This allows you to maintain journal reference information for any particular source from your feeder systems.

Descriptive Flexfield Conversion

With Journal Import, you can import up to three descriptive flexfield structures along with your journal entry lines. You can also specify whether you want Journal Import to validate your descriptive flexfields.

Government Transaction Code Conversion

With Journal Import, you can import government transaction codes to create additional debit and credit pairs from imported journal entry lines.

Multiple Charts of Accounts

Journal Import supports multiple charts of accounts. With Journal Import, you can import accounting data which applies to multiple sets of books with different charts of accounts.

**Oracle Government
General Ledger Only**

Foreign Currency Journal Entries

With Journal Import, you can import all your foreign currency data. Journal Import automatically creates foreign currency journal entries from your imported foreign currency data.

Intercompany Journal Entries

With Journal Import, you can import all your intercompany data. Once you import your intercompany data, Journal Import creates intercompany journal entries for you. And, if you want, your Oracle General Ledger application automatically balances your intercompany journal entries during posting to an intercompany account you specify.

Statistical Journal Entries

With Journal Import, you can import all your statistical data. Journal Import automatically creates statistical journal entries from your imported statistical data.

Sequential Numbering

With Journal Import, you can import sequence numbers for detail transactions from your feeder systems. In addition, you can choose to automatically assign sequential numbers to the journal entries created by Journal Import to comply with audit requirements of your country.

Budget and Encumbrance Journal Entries

With Journal Import, you can import all your budget and encumbrance data. Journal Import automatically creates budget and encumbrance entries from your imported data.

Journal Import Definitions

To understand how Journal Import works, you should be familiar with the following definitions:

Accounting Flexfield Structure

Your Oracle General Ledger application account code structure which you define to fit your specific accounting needs.

Accounting Flexfield Segment

One element of your account code, such as company or fund, division, region or product. Each Accounting Flexfield can contain up to 30 segments, and each segment can contain up to 25 characters.

Accounting Flexfield Segment Value

A series of characters and a description that define a unique value for an Accounting Flexfield segment. For example 1000, Cash Account or 510, Manufacturing Cost Center.

Set of Books

A company or fund within Oracle Financials that share a common chart of accounts structure, calendar, and functional currency. A set of books can also identify a group of companies or funds.

Journal Entry Source

Source by which your Oracle General Ledger application identifies and differentiates the origin of your journal entries.

Journal Entry Category

Category your Oracle General Ledger application uses to describe the purpose or type of journal entry.

Import Program

A custom program you use to import your detail accounting transactions from an external system into your Oracle General Ledger application. The type of loader program you write depends on the environment from which you are importing data.

Sequential Number

A number that is automatically or manually assigned to your journal entries to comply with audit requirements of your country. You assign sequences to your journal entries according to application, set of books, journal category and journal entry creation method.

Prerequisites

To ensure that your initial run of Journal Import goes smoothly, you should perform the following steps:

- Run the Optimizer to create indexes on your Accounting Flexfield segments
- Configure the Journal Import program to maximize use of your computer memory and disk space
- Define any Accounting Flexfield combinations

Journal Import runs much faster when it does not have to create new Accounting Flexfield combinations dynamically.



Overview of Setting Up
Overview of Journal Entry
(*Oracle General Ledger Reference Manual* or *Oracle Government General Ledger Reference Manual*)

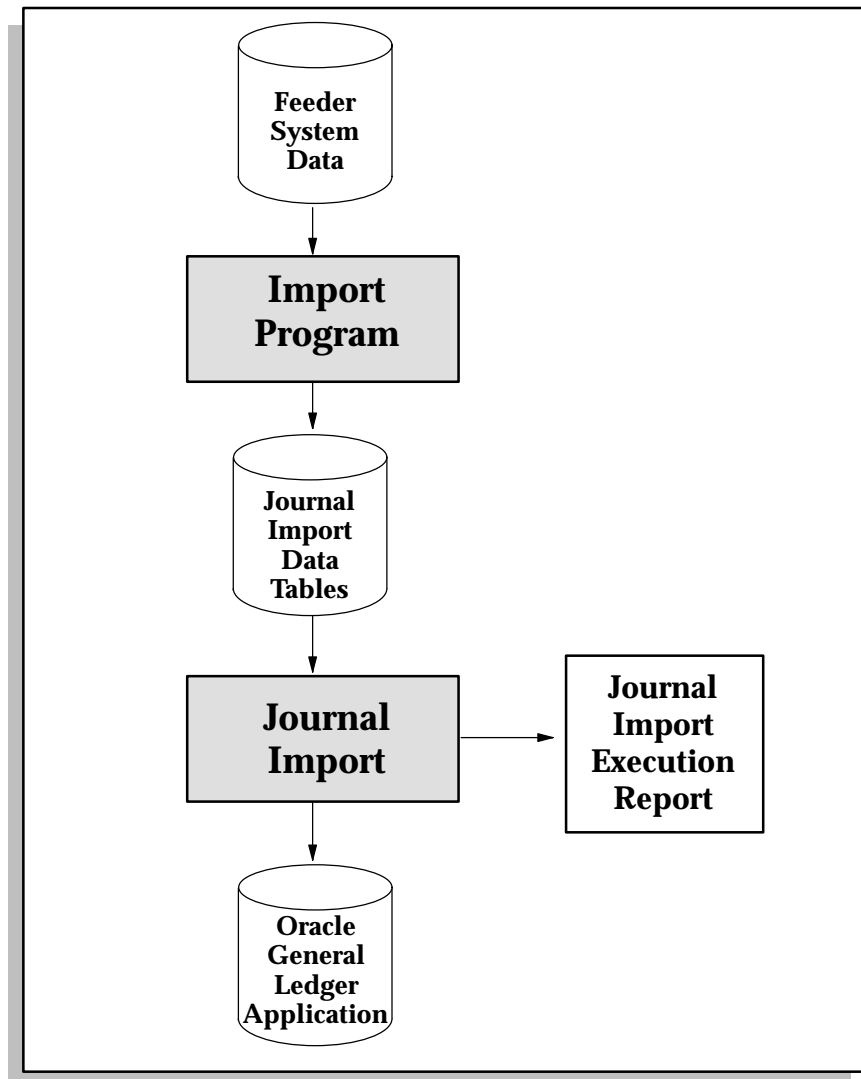


Run Optimizer
Define Concurrent Program Controls
Define Accounting Flexfield Combination
(*Oracle General Ledger Reference Manual* or *Oracle Government General Ledger Reference Manual*)

Importing Data From Your Feeder System

Journal Import receives data from an import program which your on-site MIS personnel can develop for you. Or, you may want to use an Oracle consultant. The goal of your import program is to convert data from your feeder system into a standard data format that Journal Import can read. Journal Import can then convert your import data into your Oracle General Ledger application journal entries.

Figure 4 – 1 Journal Import



Writing an Import Program

The type of environment from which you want to import data determines the type of import program you need to write. For example, you can use SQL*Loader to write an import program to import data from a non-Oracle system. Or, you can write an import program to import historical data from your previous accounting system. Regardless of the type of import program you write, the output should be in standard data format that Journal Import can use

to convert your import data into your Oracle General Ledger application journal entries.

Selecting an Import Utility

SQL*Loader is a powerful and easy-to-use tool you can use to import your accounting data. SQL*Loader should be able to accommodate all of your import needs. Depending on the complexity of your import program, there is a variety of programming languages available to you. You can use Pro-C language products such as C, Cobol and Fortran to write your import program.

Understanding the GL_INTERFACE Table

The GL_INTERFACE table is where Journal Import receives accounting data that you import from other systems. When Journal Import receives this data, it validates and converts your import data into journal entries within your Oracle General Ledger application. The GL_INTERFACE table is organized by columns in which your Oracle General Ledger application categorizes and stores specific accounting data. For example, journal entry source information is stored in the column called JE_SOURCE_NAME. The GL_INTERFACE table contains the following columns:

Column Name	Null?	Type
STATUS	NOT NULL	VARCHAR2 (50)
SET_OF_BOOKS_ID	NOT NULL	NUMBER (15)
USER_JE_SOURCE_NAME	NOT NULL	VARCHAR2 (25)
USER_JE_CATEGORY_NAME	NOT NULL	VARCHAR2 (25)
ACCOUNTING_DATE	NOT NULL	DATE
CURRENCY_CODE	NOT NULL	VARCHAR2 (15)
DATE_CREATED	NOT NULL	DATE
CREATED_BY	NOT NULL	NUMBER (15)
ACTUAL_FLAG	NOT NULL	VARCHAR2 (1)
ENCUMBRANCE_TYPE_ID		NUMBER
BUDGET_VERSION_ID		NUMBER

Table 4 – 1 GL_INTERFACE Table (Page 1 of 3)

Column Name	Null?	Type
CURRENCY_		DATE
CONVERSION_DATE		
USER_CURRENCY_		VARCHAR2 (30)
CONVERSION_TYPE		
CURRENCY_		NUMBER
CONVERSION_RATE		
SEGMENT1 through SEGMENT30		VARCHAR (25)
ENTERED_DR		NUMBER
ENTERED_CR		NUMBER
ACCOUNTED_DR		NUMBER
ACCOUNTED_CR		NUMBER
TRANSACTION_DATE		DATE
REFERENCE1		VARCHAR2 (100)
REFERENCE2		VARCHAR2 (240)
REFERENCE3		VARCHAR2 (100)
REFERENCE4		VARCHAR2 (100)
REFERENCE5		VARCHAR2 (240)
REFERENCE6 through REFERENCE9		VARCHAR2 (100)
REFERENCE10		VARCHAR2 (240)
REFERENCE11 through REFERENCE20		VARCHAR2 (100)
REFERENCE21 through REFERENCE30		VARCHAR2 (240)
GROUP_ID		NUMBER (15)
JE_BATCH_ID		NUMBER (15)
PERIOD_NAME		VARCHAR2 (15)
JE_HEADER_ID		NUMBER (15)
JE_LINE_NUM		NUMBER (15)
CHART_OF_ACCOUNTS_ID		NUMBER (15)

Table 4 – 1 GL_INTERFACE Table (Page 2 of 3)

Column Name	Null?	Type
FUNCTIONAL_CURRENCY_CODE		VARCHAR2 (15)
CODE_COMBINATION_ID		NUMBER (15)
DATE_CREATED_IN_GL		DATE
WARNING_CODE		VARCHAR2 (4)
STATUS_DESCRIPTION		VARCHAR2 (240)
DESCR_FLEX_ERROR_MESSAGE		VARCHAR2 (240)
STAT_AMOUNT		NUMBER
REQUEST_ID		NUMBER (15)
SUBLEDGER_DOC_SEQUENCE_ID		NUMBER
SUBLEDGER_DOC_SEQUENCE_VALUE		NUMBER
USSGL_TRANSACTION_CODE		VARCHAR2 (30)
ATTRIBUTE1 through ATTRIBUTE20		VARCHAR2 (150)
CONTEXT		VARCHAR2 (150)
CONTEXT2		VARCHAR2 (150)
CONTEXT3		VARCHAR2 (150)
INVOICE_DATE		DATE
INVOICE_AMOUNT		NUMBER
INVOICE_IDENTIFIER		VARCHAR2 (20)
TAX_CODE		VARCHAR2 (15)

Table 4 – 1 GL_INTERFACE Table (Page 3 of 3)

Assigning Values for Accounting Flexfields

You can specify your Accounting Flexfields in the GL_INTERFACE table in one of two ways: segment specification or code combination ID specification.

Segment Specification

Assign an Accounting Flexfield value for each segment that you enabled in your Oracle General Ledger application. For example, if you enabled four Accounting Flexfield segments, you need to first

determine into which columns of the GL_INTERFACE table you should enter data, since the SEGMENTn columns do not necessarily correspond one to one to the segments you see in the flexfield window (see the example below). This can be done by looking at the Column field of each segment in the Define Key Flexfield Segments form.

For example:

Segment 1 corresponds to the SEGMENT1 column,
Segment 2 corresponds to the SEGMENT2 column,
Segment 3 corresponds to the SEGMENT4 column, and
Segment 4 corresponds to the SEGMENT5 column (SEGMENT3 column is not used).

Given the above information above, you should load the data as follows:

data for the flexfield Segment 1 into GL_INTERFACE.SEGMENT1,
data for the flexfield Segment 2 into GL_INTERFACE.SEGMENT2,
data for the flexfield segment 3 into GL_INTERFACE.SEGMENT4, and
data for the flexfield segment 4 into GL_INTERFACE.SEGMENT5.

Load valid enabled segment values for your enabled segments into the GL_INTERFACE table. The segment values must already be defined in your Oracle General Ledger application.

For example, value 01 is not the same as value 1. You can specify Maximum Size and Right-justify Zero-fill Numbers when you define your value sets in the Define Value Set form. Maximum Size indicates the maximum width of each segment value that Journal Import expects. Right-justify Zero-fill Numbers indicates whether your Accounting Flexfield should right justify and zero-fill numbers when you enter values for a particular value set. If you have Right-justify Zero-fill Numbers set to Yes and your Maximum Size is three, then your segment value would be 001. However, if your Maximum Size is four, then your segment value would be 0001. Journal Import does not allow null values in enabled segments.

Code Combination ID Specification

Alternatively, you can enter a code combination ID to identify your Accounting Flexfield segments. You can find a list of valid Accounting Flexfield code combinations and their corresponding code combination IDs in the GL_CODE_COMBINATIONS table. If you want Journal Import to use the code combination ID to create your journal entries, enter the appropriate code combination ID in the CODE_COMBINATION_ID column of the GL_INTERFACE table and do not enter values in the SEGMENT1 through SEGMENT30 columns.

If you enter values for your Accounting Flexfield segments in the SEGMENT1 through SEGMENT30 columns and enter a value in the CODE_COMBINATION_ID column, Journal Import uses the Segment column values to create your journal entries.

If you enter segment values for your Accounting Flexfield in the GL_INTERFACE table, Oracle Applications prints the invalid Accounting Flexfield in your Journal Import Execution Report. If you enter a code combination ID and if suspense posting is disabled, Oracle Applications prints the invalid code combination ID in your Journal Import Execution Report. If you enter a code combination ID and if suspense posting is enabled, Oracle Applications prints only the segment value separators in your Journal Import Execution Report. Therefore, we recommend that you disable suspense posting if entering code combination IDs.



Overview of Setting Up
(Oracle General Ledger Reference Manual or Oracle Government General Ledger Reference Manual)

Assigning Values for Additional Required and Conditionally Required Columns

You must enter values in all columns of the GL_INTERFACE table that require values, which includes all of the not null columns, in order for Journal Import to successfully convert your import data into journal entries.

Enter values in the following required columns of the GL_INTERFACE table:

<u>Column Name</u>	<u>Value</u>
STATUS	Enter the value NEW to indicate that you are bringing new data into your Oracle General Ledger application.
SET_OF_BOOKS_ID	Enter the appropriate set of books ID for your transaction. You define your set of books in the Define Set of Books form of your Oracle General Ledger application. You can find a list of valid values in the SET_OF_BOOKS_ID column of the Sets of Books table (GL_SETS_OF_BOOKS.SET_OF_BOOKS_ID).



Suggestion: You may use the following SQL*Statement to access the appropriate set of books ID:

```
SELECT SET_OF_BOOKS_ID, NAME
FROM GL_SETS_OF_BOOKS;
```

USER_JE_SOURCE_NAME	Enter the journal entry source name for your transaction. You define journal sources in the Define Journal Entry Sources form of your Oracle General Ledger application. You can find a list of valid values in the USER_JE_SOURCE_NAME column of the Journal Entry Sources table (GL_JE_SOURCES.USER_JE_SOURCE_NAME).
USER_JE_CATEGORY_NAME	Enter the journal category name for your transaction. You define journal categories in the Define Journal Entry Categories form of your Oracle General Ledger application. You can find a list of valid values in the USER_JE_CATEGORY_NAME column of the Journal Entry Categories table (GL_JE_CATEGORIES.USER_JE_CATEGORY_NAME).
ACCOUNTING_DATE	Enter the accounting date on which your transaction occurred. Your Oracle General Ledger application automatically assigns your journal batch to the earliest accounting period that includes your accounting date.
CURRENCY_CODE	Enter the currency code for your transaction. You define new currency codes in the Define Currency form of your Oracle General Ledger application. You can find a list of valid values in the CURRENCY_CODE column of the Currencies table (FND_CURRENCIES.CURRENCY_CODE).
DATE_CREATED	Enter the date your import journal entry line was created. The information you enter here is for your own records, and does not appear in your Oracle General Ledger application.
CREATED_BY	Enter an ID that you can use to identify the data coming from your feeder system. The ID you enter provides you with an audit trail from Journal Import data to your feeder system. However, your Journal Import data will be removed from the GL_INTERFACE table after it is successfully imported, and this ID will not appear in your Oracle General Ledger application.

ACTUAL_FLAG	Enter the value A for actual amounts, B for Budget amounts, or E for encumbrance amounts.
ENCUMBRANCE_TYPE_ID	If you entered the value E in the ACTUAL_FLAG column of the GL_INTERFACE table, you must enter the appropriate encumbrance ID. You define new encumbrance types in the Define Encumbrance Types form of your Oracle General Ledger application. You can find a list of valid values in the ENCUMBRANCE_TYPE_ID column of the Encumbrance Types table (GL_ENCUMBRANCE_TYPES.ENCUMBRANCE_TYPE_ID).



Suggestion: We recommend you use the following SQL*Statement to identify the appropriate encumbrance type ID:

```
SELECT ENCUMBRANCE_TYPE_ID,
ENCUMBRANCE_TYPE
FROM GL_ENCUMBRANCE_TYPES
WHERE ENABLED_FLAG = 'Y';
```

BUDGET_VERSION_ID	If you entered the value B in the ACTUAL_FLAG column of the GL_INTERFACE table, you must enter the appropriate budget ID. You define new budget versions in the Define Budget form of your Oracle General Ledger application. You can find a list of valid values in the BUDGET_VERSION_ID column of the Budget Versions table (GL_BUDGET_VERSIONS.BUDGET_VERSION_ID).
--------------------------	--



Suggestion: We recommend you use the following SQL*Statement to identify the appropriate budget version ID:

```
SELECT BUDGET_VERSION_ID,
BUDGET_NAME
FROM GL_BUDGET_VERSIONS
WHERE STATUS IN ('C','O');
```

PERIOD_NAME	Enter a period name for your budget transactions (ACTUAL_FLAG = B) only. This column is
--------------------	---

required when you are importing budget data using Journal Import. If you want to import budget data using Journal Import, you must supply a period name instead of an accounting date. And, your period name must be associated with an open budget fiscal year.

ENTERED_DR Enter the debit amount for each line of your transaction. Enter a value for the ENTERED_DR or the ENTERED_CR column in a given row, but not both values in one row.

ENTERED_CR Enter the credit amount for each line of your transaction. Enter a value for the ENTERED_DR or the ENTERED_CR column in a given row, but not both values in one row.



- Define Set of Books
- Define Budget
- Define Currency
- Define Encumbrance Types
- Define Journal Entry Sources
- Define Journal Entry Categories
- (Oracle General Ledger Reference Manual or Oracle Government General Ledger Reference Manual)*

Assigning Values for Currency Conversion

You can enter values for your actual foreign currency data in one of two ways. You can specify the entered amount along with a conversion rate type and date and let your Oracle General Ledger application calculate the converted amount for you. Or, you can directly specify the entered and converted amounts and not specify the conversion rate, type and date.

Do not enter values in the following columns for encumbrance and budget foreign currency data. Enter values for your actual foreign currency data only in the following columns of the GL_INTERFACE table:

	<u>Column Name</u>	<u>Value</u>
System Calculated Conversion	USER_CURRENCY_CONVERSION_TYPE	Enter a currency conversion <i>type</i> for your actual foreign currency transactions. Acceptable values are User, Spot, Corporate, or any other type you define in the Define Daily Conversion Rate Types form. If you enter a rate type of User, then you must also enter a conversion rate in the

		<p>CURRENCY_CONVERSION_ RATE column. For all other conversion types you must enter a conversion date in the CURRENCY_CONVERSION_ DATE column.</p> <p>You can find a list of valid values in the USER_CURRENCY_CONVERSION_TYPE column of the Conversion Types table (GL_DAILY_CONVERSION_TYPES.USER_CURRENCY_CONVERSION_TYPE).</p>
	CURRENCY_CONVERSION_ DATE	<p>Enter a currency conversion <i>date</i> for your actual foreign currency transactions. If you enter a conversion type other than User in the USER_CURRENCY_CONVERSION_TYPE column, you must enter a value in this column. If your conversion type is User, the default value for this column is the accounting date.</p>
	CURRENCY_CONVERSION_ RATE	<p>Enter a currency conversion <i>rate</i> for your actual foreign currency transactions. If you enter a conversion type of User in the USER_CURRENCY_CONVERSION_TYPE column, you must enter a value in this column. If you enter a conversion type other than USER, do not enter anything in this column.</p>
User Entered Conversion	ACCOUNTED_ DR	<p>Enter a converted debit amount for your actual foreign currency transactions. Enter a value for the ACCOUNTED_DR or the ACCOUNTED_CR column in a given row, but not both values in one row. You must enter a value for ENTERED_DR if you entered a value for ACCOUNTED_DR.</p>
	ACCOUNTED_ CR	<p>Enter a converted credit amount for your actual foreign currency transactions. Enter a value for the ACCOUNTED_DR or the ACCOUNTED_CR column in a given row, but not both values in one row. You must enter a value for ENTERED_CR if you entered a value for ACCOUNTED_CR.</p>

Assigning Values to Optional Columns

You can enter values for many optional columns in the GL_INTERFACE table. Enter values in these columns for maximum control over the way Journal Import groups the journal entry lines it creates into journal entries. If you do not enter a value in an optional

column and a default value exists for that particular column, Journal Import automatically enters the default value.

Enter values in the following optional columns of the GL_INTERFACE table:

<u>Column Name</u>	<u>Value</u>
REFERENCE1 (Batch Name)	Enter a batch name for your import batch. Journal Import creates a default batch name using the following format: (Optional User-Entered REFERENCE1) (Source) (Request ID) (Actual Flag) (Group ID). If you enter a batch name, Journal Import prepends the first 50 characters of your batch name to the above format.
REFERENCE2 (Batch Description)	Enter a description for your batch. If you do not enter a batch description, Journal Import automatically gives your batch a description using the format: Journal Import (Source) (Request Id).
REFERENCE4 (Journal entry name)	Enter a journal entry name for your journal entry. Journal Import creates a default journal entry name using the following format: (Category Name) (Currency) (Currency Conversion Type, if applicable) (Currency Conversion Rate, if applicable) (Currency Conversion Date, if applicable) (Encumbrance Type ID, if applicable) (Budget Version ID, if applicable). If you enter a journal entry name, Journal Import prepends the first 25 characters of your journal entry name to the above format.
REFERENCE5 (Journal entry description)	Enter a description for your journal entry. If you do not enter a journal entry description, Journal Import automatically gives your journal entry a description using the format: Journal Import – Concurrent Request ID.
REFERENCE6 (Journal entry reference)	Enter a reference name or number for your journal entry. If you do not enter a journal entry reference, Journal Import automatically creates a journal entry reference called Journal Import Created.
REFERENCE7 (Journal entry reversal flag)	Enter Yes to reverse your journal entry. If you do not enter Yes, Journal Import automatically defaults to No.

REFERENCE8 (Journal entry reversal period)	Enter the name of the period to which you want to reverse your journal entry. If you enter Yes in the REFERENCE7 column, you must enter a value in this column.
REFERENCE10 (Journal entry line description)	Enter a description for your journal entry line. If you do not enter a journal entry line description, Journal Import uses the subledger document sequence value. If there is no document sequence value, Journal Import creates a journal entry description called Journal Import Created.
REFERENCE21 through REFERENCE30	<p>Enter a reference name or number to further identify your import journal entry lines. Columns REFERENCE21 through REFERENCE30 map into columns REFERENCE_1 through REFERENCE_10, respectively, of the GL_JE_LINES table.</p> <p>Once in the GL_JE_LINES table, your Oracle General Ledger application prints the value stored in REFERENCE_1 in standard reports run with Line detail, and prints the value stored in REFERENCE_4 in standard reports run with Source detail. The other reference columns are for descriptive or tracking purposes only. The values in these columns are not used in your Oracle General Ledger application.</p>
GROUP_ID	Enter a unique group number to distinguish import data within a source. You can run Journal Import in parallel for the same source if you specify a unique group number for each request.
STAT_AMOUNT	Enter the statistical amount associated with your journal entry line data. You define statistical units of measure in the Define Statistical Units of Measure form of your Oracle General Ledger application. You must use this column when you want to see statistical and monetary amounts in the same journal entry line.
USSGL_TRANSACTION_CODE	Enter a valid USSGL transaction code for your journal entry line. Journal Import validates and imports the USSGL transaction codes when you have the profile option Government: Enable Transaction Code set to Yes, and you have defined your USSGL transaction codes using the Define Government Transaction Codes form.

ATTRIBUTE1 through ATTRIBUTE 10	Enter values for your descriptive flexfield "Journals – Journal Entry Line". The values you enter depend on how you defined your descriptive flexfield in the Define Descriptive Flexfield Segments form.
ATTRIBUTE11 through ATTRIBUTE 20	Enter values for your descriptive flexfield "Journals – Captured Information". The values you enter depend on how you defined your descriptive flexfield in the Define Descriptive Flexfield Segments form.
CONTEXT	Enter the context field value for the descriptive flexfield "Journals – Journal Entry Line" that identifies the structure of your descriptive flexfield. If you enter a value, you can also enter some combination of values in the columns ATTRIBUTE1 through ATTRIBUTE10.
CONTEXT2	Enter Yes to identify your Value Added Tax Descriptive Flexfield structure. You must use this column if you import data for the Value Added Tax Descriptive Flexfield. Enter No to indicate that your journal entry line is not a tax item. If you enter No, the four Value Added Tax Descriptive Flexfield related columns must be null.
CONTEXT3	Enter the context field value (natural account) for the descriptive flexfield "Journals – Captured Information" that identifies the structure of your descriptive flexfield. Enter a value only if you are importing the descriptive flexfield "Journals – Captured Information" without validation. If you enter a value, you can also enter some combination of values in the columns ATTRIBUTE11 through ATTRIBUTE20.
INVOICE_DATE	Enter the date on which you paid or collected tax on your tax journal entry line. Enter the date in the format DD-MON-YY or the default date format for your language. Your invoice date should correspond to the date when tax amounts were paid or received for this invoice. You must use this column if you import data for the Value Added Tax Descriptive Flexfield.
INVOICE_ AMOUNT	Enter an invoice amount. Enter the net invoice amount that relates to your tax journal entry line

	amount. You must use this column if you import data for the Value Added Tax Descriptive Flexfield.
TAX_CODE	Enter a valid tax code that identifies the type of tax paid for this invoice. You define a list of valid tax codes for this field when you define your descriptive flexfield values. You must use this column if you import data for the Value Added Tax Descriptive Flexfield.
INVOICE_IDENTIFIER	Enter an invoice identifier. Enter reference information about the source document or invoice upon which you paid or collected tax. You must use this column if you import data for the Value Added Tax Descriptive Flexfield.

Required NULL Columns in the GL_INTERFACE Table

Some columns in the GL_INTERFACE table must be NULL as Journal Import uses them for internal processing or does not use them in the current release. The following columns must be NULL in release 10 of your Oracle General Ledger application:

<u>Column Name</u>	<u>Value</u>
REFERENCE3	Do not enter a value in this column.
REFERENCE9	Do not enter values in this column.
REFERENCE11 through REFERENCE20	Do not enter a value in this column.
TRANSACTION_DATE	Do not enter a value in this column.
JE_BATCH_ID	Do not enter a value in this column.
JE_HEADER_ID	Do not enter a value in this column.
JE_LINE_NUM	Do not enter a value in this column.
CHART_OF_ACCOUNTS_ID	Do not enter a value in this column.
FUNCTIONAL_CURRENCY_CODE	Do not enter a value in this column.
DATE_CREATED_IN_GL	Do not enter a value in this column.

WARNING_ CODE	Do not enter a value in this column.
STATUS_ DESCRIPTION	Do not enter a value in this column.
DESC_FLEX_ ERROR_ MESSAGE	Do not enter a value in this column.
REQUEST_ID	Do not enter a value in this column.
SUBLEDGER_ DOC_ SEQUENCE_ID	Do not enter a value in this column.
SUBLEDGER_ DOC_ SEQUENCE_ VALUE	Used for communication between Oracle General Ledger and the subledgers. Do not populate with your own data.



Overview of Journal Entry
Overview of Multi-Currency
(*Oracle General Ledger Reference Manual* or *Oracle Government General Ledger Reference Manual*)



Define Daily Conversion Rate Types
Define Government Transaction Codes
Define Statistical Units of Measure
(*Oracle General Ledger Reference Manual* or *Oracle Government General Ledger Reference Manual*)



Define Descriptive Flexfield Segments
Define Key Segment Values
(*Oracle Applications Flexfields Manual*)

Importing Specialized Data

Multi-Currency Data

Load multi-currency data into the GL_INTERFACE table the same way you load regular data. If you want your Oracle General Ledger application to calculate your conversion, you must enter a value in the CURRENCY_CODE, CURRENCY_CONVERSION_DATE and

USER_CURRENCY_CONVERSION_TYPE columns of the GL_INTERFACE table. If the conversion type is User, you must also enter a value in the CURRENCY_CONVERSION_RATE column of the GL_INTERFACE table. Or, you can directly specify the converted amounts by entering values in the ACCOUNTED_DR and ACCOUNTED_CR columns. If you choose to enter your converted amounts, do not specify the conversion rate, type and date.

Intercompany or Interfund Data

Load intercompany or interfund data into the GL_INTERFACE table the same way you load regular data. Journal Import creates intercompany or interfund journal entries from the data you import. And, if you want, your Oracle General Ledger application automatically balances your intercompany or interfund journal entries during posting to an intercompany or interfund account you specify when you define your set of books.

Statistical Data

Load statistical data into the GL_INTERFACE table the same way you load regular data. The only difference is that you enter the value STAT in the CURRENCY_CODE column of the GL_INTERFACE table. Do not enter values in the STAT_AMOUNT column.

Alternatively, if you choose to use units of measure, you can enter a positive amount for a debit or a negative amount for a credit in the STAT_AMOUNT column of the GL_INTERFACE table for each monetary journal entry line amount. In this case, enter a monetary currency, not STAT, in the CURRENCY_CODE column.

Encumbrance Data

Load encumbrance data into the GL_INTERFACE table the same way you load regular data. The only difference is that you must enter the value E in the ACTUAL_FLAG column and the appropriate encumbrance type ID in the ENCUMBRANCE_TYPE_ID column of the GL_INTERFACE table.

Budget Data

Load budget data into the GL_INTERFACE table the same way you load regular data. The only difference is that you must enter the value B in the ACTUAL_FLAG column and the appropriate budget version ID in the BUDGET_VERSION_ID column of the GL_INTERFACE table.

You must enter a valid period name for budget journal batches created by Journal Import. Use the PERIOD_NAME column to enter a valid batch period whenever you specify the value B in the ACTUAL_FLAG column of the GL_INTERFACE table.

Journal Import Validation

Journal Import validates all of your data before it creates journal entries in your Oracle General Ledger application. If you allow suspense posting for your set of books, Journal Import will assign lines with invalid Accounting Flexfields to that account. Journal Import rejects all other invalid lines, and they remain in the GL_INTERFACE table where you can correct them online in the Correct Journal Import Data form or in your feeder system. Journal Import also prints your error lines in the Journal Import Execution Report.

Batch Level Validation

Journal Import validates the following attributes to ensure that a batch with the same name does not already exist for the same set of books and period in your Oracle General Ledger application:

- Set of books
- Period name
- Batch name

Journal Import also checks to ensure that more than one journal entry with the same name does not exist for a batch.

Journal Entry Line Level Validation

Journal Import validates the following attributes to ensure that your journal entry lines contain the appropriate accounting data:

- Set of books
- Period name
- Batch name
- Source name
- Journal entry name
- Currency code
- Category name

- Actual flag
- Encumbrance type ID
- User conversion type
- Accounting date
- Budget version ID
- Reversal period (GL_INTERFACE.REFERENCE8)
- USSGL Transaction Code

Accounting Flexfield Validation

Journal Import validates your Accounting Flexfield code combinations in a number of ways. Journal Import will successfully import your accounting data if your code combinations meet the following validation requirements:

- You allow detail posting to segment combinations.
- You have enabled your code combinations for the accounting date you specify.
- Your code combinations do not include summary accounts.

Descriptive Flexfield Validation

Journal Import validates your descriptive flexfield segments in a number of ways depending on the particular descriptive flexfield. If your descriptive flexfield segments are null, then Journal Import does not validate the descriptive flexfield. Otherwise, Journal Import will successfully import your descriptive flexfield data if your descriptive flexfield segments meet the following validation requirements:

Journals – Journal Entry Line Descriptive Flexfield

- The descriptive flexfield global segments have valid values.
- The descriptive flexfield context is a valid value.
- The descriptive flexfield context dependent segments have valid values.

Journals – Captured Information Descriptive Flexfield

- The descriptive flexfield global segments have valid values.
- The descriptive flexfield context dependent segments have valid values.

Value Added Tax Descriptive Flexfield

- The descriptive flexfield context is set to Yes or No.
- The descriptive flexfield context dependent segments have valid values.

Preparing your Oracle General Ledger application

Before using Journal Import, you should prepare your Oracle General Ledger application for any new data that you want Journal Import to import.

Accounting Flexfield Segment Values

Add Accounting Flexfield segment values to your Oracle General Ledger application if your feeder systems include values not yet defined in your Oracle General Ledger application.



Define Key Segment Values
(*Oracle Applications Flexfields Manual*)

Sets of Books

Add sets of books to your Oracle General Ledger application if your feeder systems include sets of books not yet defined in your Oracle General Ledger application.



Define Set of Books
(*Oracle General Ledger Reference Manual* or *Oracle Government General Ledger Reference Manual*)

Currencies

Add currencies to your Oracle General Ledger application if your feeder systems include currencies not yet defined in your Oracle General Ledger application.



Define Currency
(*Oracle General Ledger Reference Manual* or *Oracle Government General Ledger Reference Manual*)

Journal Entry Sources

Add journal entry sources to your Oracle General Ledger application if your feeder systems include journal entry sources not yet defined in your Oracle General Ledger application.

You can also specify whether you want your Oracle General Ledger to store journal reference information from your feeder systems for a particular source. This feature is useful when you want to maintain a mapping of how your detail transactions from your feeder systems are summarized into journal lines by Journal Import when you choose to create summary journals. Journal Import stores mapping information in the GL_IMPORT_REFERENCES table. You must write your own report referencing the table.



Define Journal Entry Sources

(Oracle General Ledger Reference Manual or Oracle Government General Ledger Reference Manual)

Journal Entry Categories

Add journal entry categories to your Oracle General Ledger application if your feeder systems include categories not yet defined in your Oracle General Ledger application.



Define Journal Entry Categories

(Oracle General Ledger Reference Manual or Oracle Government General Ledger Reference Manual)

Sequential Numbering

Your Oracle General Ledger application allows you to generate a sequence number for every journal entry created by Journal Import if you enable sequential numbering. If you want Journal Import to assign sequential numbers to your journal entries, you must specify Automatic as both your numbering and document generation method.



Define Document Sequences

Assign Document Sequences

(Oracle General Ledger Reference Manual or Oracle Government General Ledger Reference Manual)

Government Transaction Codes

Oracle Government
General Ledger Only

Add government transaction codes to Oracle Government General Ledger if your feeder systems include journal entry government transaction codes not yet defined in your Oracle Government General Ledger.



Define Government Transaction Codes

(Oracle Government General Ledger Reference Manual)

Using Journal Import

Defining Concurrent Program Controls

You must run the Optimizer before defining your concurrent program controls. Once done, you can improve the performance of Journal Import (and MassAllocations) by varying the amount of disk space and memory it uses. The setup options you define in the Define Concurrent Program Controls form are used as a default whenever you run Journal Import or MassAllocations. Once defined, you will rarely change the settings except if you add or remove memory.

Your Oracle General Ledger application reference manual explains how to define default settings that optimize the memory and disk space available in your computer. You need to know how much memory and disk space you want to allocate for concurrent programs before using the Define Concurrent Program Controls form. The Journal Import and MassAllocations programs each require approximately 1.4 megabytes of memory to run.

You can also specify whether to save your Journal Import data each time you run Journal Import. Journal Import runs faster if you do not archive your data.



Run Optimizer

Define Concurrent Program Controls

(Oracle General Ledger Reference Manual or Oracle Government General Ledger Reference Manual)

Running Journal Import

Run Journal Import using the Run Journal Import form in your Oracle General Ledger application. You enter the sources from which you want to import accounting data. Your Oracle General Ledger application lets you import transactions whenever you want, and you can update or post them just like any other journal entry.



Run Journal Import

(Oracle General Ledger Reference Manual or Oracle Government General Ledger Reference Manual)

Running Journal Import can be a one-step or multi-step process depending on the accuracy of your import program. If your import program converts your journal entries from other sources into the required data format, and all of the data is valid in your Oracle General Ledger application, then Journal Import should run successfully the first time. However, if you load data into the GL_INTERFACE table

which is not valid in your Oracle General Ledger application, Journal Import informs you of the specific errors on the Journal Import Execution Report. In this case, you can correct your errors online using the Correct Journal Import Data form. Or, you can delete your import data from the GL_INTERFACE table using the Delete Source From Journal Import form, correct the errors and then re-import your data.



Correct Journal Import Data
Delete Source from Journal Import
(*Oracle General Ledger Reference Manual* or *Oracle Government General Ledger Reference Manual*)

Reviewing the Journal Import Execution Report

Use the Journal Import Execution Report to review the status of all import journal entries. The Journal Import Execution Report prints a line for each journal entry source from which you are importing journal entries. Each line of the report includes the total number of journal entry lines Journal Import successfully imported. Each report line also includes any problems Journal Import may have encountered during the import process. Refer to the error key section of the report to find an explanation of the errors. You can use this information to correct your errors and successfully re-import your batch information.



Run Reports
(*Oracle General Ledger Reference Manual* or *Oracle Government General Ledger Reference Manual*)

Deleting Incorrect Data

Use the Delete Source from Journal Import form to reverse the journal import process for specific sources containing errors. Refer to the Journal Import Execution Report to determine which sources contain errors. We recommend that you make any corrections to your data from within your feeder systems to preserve your audit trail. However, you can correct errors online using the Correct Journal Import Data form.

If you decide to correct your errors in your feeder system, you should delete all records with this import source from Journal Import. Use the Delete Source from Journal Import form to delete all records from the GL_INTERFACE_TABLE, then re-import this source. To reverse the journal import process, enter the name of the source and the Run ID for

the specific set of imported journal entries that you want to delete. Obtain the Run ID from the Journal Import Execution Report.



Correct Journal Import Data

Delete Source from Journal Import

(Oracle General Ledger Reference Manual or Oracle Government General Ledger Reference Manual)

Correcting Errors Online

If you allow suspense posting, Journal Import will create suspense lines only for accounting data which violates Accounting Flexfield validation rules. If your data violates any other types of validation checks, your accounting data remains in the GL_INTERFACE table, where it must be corrected. You can correct errors based on the error code you receive on the Journal Import Execution Report. You can make changes using the Correct Journal Import Data form.



Correct Journal Import Data

(Oracle General Ledger Reference Manual or Oracle Government General Ledger Reference Manual)

Journal Import Journal Entry Creation

Once Journal Import validates your import data, it transfers the data from the GL_INTERFACE table to your Oracle General Ledger application journal entry tables. Below is a listing of the Oracle General Ledger application journal entry tables:

GL_JE_BATCHES	The table your Oracle General Ledger application uses to store journal entry batch information such as name, status, and batch debit and credit totals.
GL_JE_HEADERS	The table your Oracle General Ledger application uses to store journal entry header information such as name, category, date and currency.
GL_JE_LINES	The table your Oracle General Ledger application uses to store journal entry line information such as journal entry line number and Accounting Flexfield debit and credit amounts.

Suspense Journals Created By Journal Import

You can choose to optionally create suspense journal lines when your Journal Import data violates certain Accounting Flexfield validation rules. If you define suspense accounts for each journal source and

category in the Define Suspense Accounts form, Journal Import assigns the appropriate suspense account to unbalanced journal line amounts or journal lines containing Accounting Flexfield errors. Journal Import assigns suspense accounts based on the journal source and category for the suspense journal line.

If you allow suspense posting, Journal Import creates a suspense line for the following errors:

- EF01** This Accounting Flexfield is disabled for this accounting date.
- EF02** Detail posting is not allowed for this Accounting Flexfield.
- EF03** Disabled Accounting Flexfield.
- EF04** These segment values are not a valid Accounting Flexfield. Check your cross validation rules.
- EF05** You provided a code combination ID, but there is no Accounting Flexfield with this ID.



Run Journal Import
Define Suspense Accounts
(*Oracle General Ledger Reference Manual* or *Oracle Government General Ledger Reference Manual*)

Reviewing Journal Entries

Review the journal entries created by Journal Import before you post them to your general ledger accounts in the Enter Journals form.



Enter Journals
(*Oracle General Ledger Reference Manual* or *Oracle Government General Ledger Reference Manual*)

Posting Journal Entries

Post your Journal Import journal entries.



Post Journals
(*Oracle General Ledger Reference Manual* or *Oracle Government General Ledger Reference Manual*)

Create Summary Journals

When you create journal entries using Journal Import, you can choose to summarize debits and credits for a given Accounting Flexfield. To

create summary journals, enter Yes in the Create Summary Journals field in the Run Journal Import form.



Run Journal Import

(Oracle General Ledger Reference Manual or Oracle Government General Ledger Reference Manual)

Integrating Oracle General Ledger or Oracle Government General Ledger Using Budget Upload

Budget Upload lets you prepare and analyze your budget on a personal computer using a spreadsheet program, and then transfer your spreadsheet information into your Oracle General Ledger application. You can perform your budgeting in the environment you choose, and still maintain the integrity of your database. This section illustrates the alternative methods you can use to take advantage of the Budget Upload capabilities in your Oracle General Ledger application.

Basic Integration Needs

Budget Upload provides you with the features you need to satisfy your budget integration needs. You should be able to:

- Perform budgeting on your personal computer using a spreadsheet program. You can organize your spreadsheet any way you want.
- Transfer budget information from your spreadsheet to your database. Your financial records should reflect any new budget information in your spreadsheet.
- Transfer budget information from your database to your spreadsheet. Your spreadsheet should reflect any new budget information in your database.
- Upload budget information for new as well as existing Accounting Flexfields.

Major Features

Simple Integration

With Budget Upload, you can transfer the contents of your spreadsheet to your Oracle General Ledger application with only a few keystrokes. You can even transfer budget information from your Oracle General Ledger application to your spreadsheet.

Budgeting Flexibility

With Budget Upload, you can choose the environment you want to use for budgeting. Whether you choose to do your budgeting in a spreadsheet or directly in your Oracle General Ledger application, you can plan and control the budget process.

Complete Validation

Your Oracle General Ledger application validates all of your budget information when you integrate your spreadsheet with your Oracle General Ledger application.

Dynamic Insertion

You can upload budget information for new Accounting Flexfields that fall within the Accounting Flexfield ranges assigned to your budget organization. Your Oracle General Ledger application dynamically creates new Accounting Flexfields if budgeting is allowed and assigns them to your budget organization.

Budget Spreadsheet Upload Status Report

Each time you use Budget Upload to transfer budget information from your spreadsheet to your Oracle General Ledger application, your Oracle General Ledger application automatically prints a status report. You can use the Budget Spreadsheet Upload Status Report to review the results of your budget integration.

Automatic Update

Budget Upload automatically updates the balances of your financial records when you transfer budget information from your spreadsheet to your Oracle General Ledger application.

Complete Compatibility

Budget Upload is completely compatible with any spreadsheet that can save an ASCII file. Transfer the file to the host where Oracle is running and then use SQL*Loader to load the GL_BUDGET_INTERFACE table. Budget Upload is also compatible with networking products such as SQL*Net.

Alternately, you can use Oracle Glue or Dynamic Data Exchange to populate GL_BUDGET_INTERFACE from a PC spreadsheet.

Budget Upload Definitions

To understand how Budget Upload works, you should be familiar with the following definitions:

Budget Upload

The ability to transfer budget information from a spreadsheet to your Oracle General Ledger application. For example, with the Spreadsheet Interface you can upload budget information from your spreadsheet to your Oracle General Ledger application.

Budget Organization

An entity (department, division or other group) responsible for entering and maintaining budget data. You define budget organizations for your company or agency, then assign the appropriate Accounting Flexfields to each budget organization.

Budget Interface Table

The table that stores the information you need to use the Spreadsheet Interface. Information in the Budget Interface table is stored in columns.

Prerequisites

Before you upload your budget information from a spreadsheet to your Oracle General Ledger application, you should perform the following steps in your Oracle General Ledger application:

- Define your Accounting Flexfield structure



Define Value Sets
Define Key Flexfield Segments
Define Key Segment Values
(*Oracle Applications Flexfields Manual*)

- Define your accounting periods
- Define your budgets

- Define your budget organizations



Define Period Types

Define Calendar

Define Budget

Define Budget Organization

(Oracle General Ledger Reference Manual or Oracle Government General Ledger Reference Manual)

Creating Your Spreadsheet

Understanding the Budget Interface Table

The first step in transferring your budget data from your spreadsheet application to your Oracle General Ledger application is to load your data into your Oracle General Ledger application Budget Interface table. Once you load your budget information into the Budget Interface table, you can run Budget Upload to post your budget data into your Oracle General Ledger application.

Budget Upload uses the Budget Interface table

GL_BUDGET_INTERFACE to upload budget information from your spreadsheet. You need to become familiar with this table. The Budget Interface table is organized into columns in which your Oracle General Ledger application categorizes and stores specific budget information. For example, the name of your budget is stored in the column called Budget_Name. You must specify valid values for each of the required columns in this table to successfully complete a Budget Upload. You may specify values for the optional columns within this table. These values are validated before your Oracle General Ledger application updates budget balances. The Budget Interface table contains the following columns:

Column Name	Null?	Type
BUDGET_NAME	NOT NULL	VARCHAR2 (15)
BUDGET_ENTITY_NAME	NOT NULL	VARCHAR2 (25)
CURRENCY_CODE	NOT NULL	VARCHAR2 (15)
FISCAL_YEAR	NOT NULL	NUMBER (15)
UPDATE_LOGIC_TYPE	NOT NULL	VARCHAR2 (1)
BUDGET_ENTITY_ID		NUMBER (15)
SET_OF_BOOKS_ID		NUMBER (15)
CODE_COMBINATION_ID		NUMBER (15)
BUDGET_VERSION_ID		NUMBER (15)
PERIOD_TYPE		VARCHAR2 (15)
DR_FLAG		VARCHAR2 (1)
STATUS		VARCHAR2 (1)
ACCOUNT_TYPE		VARCHAR2 (1)
LAST_UPDATE_DATE		DATE
LAST_UPDATED_BY		NUMBER (15)
REQUEST_ID		NUMBER (15)
PERIOD1_AMOUNT through PERIOD60_AMOUNT		NUMBER
SEGMENT1 through SEGMENT30		VARCHAR2 (25)

Table 4 – 2GL_BUDGET_INTERFACE Table (Page 1 of 1)

The Contents of Your Spreadsheet

Your spreadsheet should contain all of the budget information you need for your company or agency. You can arrange your budget information in any way you want. However, before you can upload your budgets from a spreadsheet you must organize your budget information according to the structure of the Budget Interface table. Therefore, each column of your spreadsheet should correspond to a

column in the Budget Interface table. For example, your spreadsheet should contain a column called Budget_Name that corresponds to the Budget Interface table column of the same name.

Figure 4 – 2

A:B23: [M1] READY

Account Name	Account	October	November	December	January	February
Salaries	5500	50,000	50,000	50,000	50,000	50,000
Commissions	5550	5,000	5,000	5,000	5,000	5,000
Benefits	5600	0	0	0	0	500
Travel & Lodging	5610	5,000	5,000	2,000	2,000	5,000
Occupancy Expense	5700	110,000	110,000	110,000	110,000	110,000
Utility Expense	5710	10,000	10,000	10,000	10,000	10,000
Miscellaneous Expense	5800	10,000	1,000	0	0	0
Advertising Expense	5750	20,000	20,000	20,000	20,000	20,000
Total		\$210,000	\$201,000	\$197,000	\$197,000	\$200,500

INTERFACE WORKSHEET

segment3	period1_amount	period2_amount	period3_amount	period4_amount	period5_amount
5500	50,000	50,000	50,000	50,000	50,000
5550	5,000	5,000	5,000	5,000	5,000
5600	0	0	0	0	500
5610	5,000	5,000	2,000	2,000	5,000
5700	110,000	110,000	110,000	110,000	110,000
5710	10,000	10,000	10,000	10,000	10,000
5800	10,000	1,000	0	0	0
5750	20,000	20,000	20,000	20,000	20,000

budgetupld.wk3

You can organize your budget information in any way you want in the working section, and then automatically copy the information in your working section to an interface section that corresponds to the Budget Interface table.

Entering Budget Information for Required Columns

Enter information for the required columns in the Budget Interface table. Assign to the following columns the values described below:

<u>Column Name</u>	<u>Value</u>
Currency_Code	Enter the currency for your account.
Budget_Name	Enter the name of the budget to which you want to budget amounts. You can only upload budgets for current or open budgets.
Budget_Entity_Name	Enter the budget organization to which you want to budget amounts.
Fiscal_Year	Enter the fiscal year to which you want to budget. You can only upload budget amounts for open budget fiscal years within a budget. You can open a new budget fiscal year in the Define Budgets form.

Update_Logic_ Type	Enter the value R or A. Enter the value R if you want the amounts you enter to replace existing amounts. Enter the value A if you want the amounts you enter to add to existing amounts.
Period1_Amount through Period60_ Amount	Enter an amount for each budget period in your open fiscal year. You can enter amounts for up to 60 periods for each open fiscal year.
Segment1 through Segment30	Enter existing or new valid Accounting Flexfield segment values in your spreadsheet for each Accounting Flexfield segment you enabled in your Oracle General Ledger application.



Define Budget
(*Oracle General Ledger Reference Manual or Oracle Government General Ledger Reference Manual*)

Entering Period Amounts

Assign a column of your spreadsheet for each budget period in your Oracle General Ledger application. You can enter amounts for up to 60 periods per fiscal year. Budget Upload assumes that you enter period amounts for each account in the sequence that the period falls in the fiscal year. For example, if your fiscal year ranges from January to December and you want to upload budget amounts for the months April, May and June, then you need to enter amounts for columns Period4_Amount through Period6_Amount.

When you enter a period amount for an Accounting Flexfield, Budget Upload updates the account balance according to the update logic you specify. If you enter R (Replace), Budget Upload replaces the existing account balance with the period amount you specify. If you enter A (Add), Budget Upload adds the period amount to the existing account balance. If you do not enter a period amount for an Accounting Flexfield, your Oracle General Ledger application does not update the account balance, regardless of update logic.

Entering Data for Accounting Flexfields

Enter the Accounting Flexfields for which you want to upload budget information from your spreadsheet. You can upload budget amounts to your budget organization for:

- An existing Accounting Flexfield that falls within the Accounting Flexfield ranges assigned to your budget organization.

- A new Accounting Flexfield that falls within the Accounting Flexfield ranges assigned to your budget organization.

Your Oracle General Ledger application dynamically creates a new Accounting Flexfield if budgeting is allowed and assigns it to your budget organization. Your Oracle General Ledger application then uploads the budget data and updates the budget account balance.

If you upload budget amounts to your budget organization for a new Accounting Flexfield that does not fall within the Accounting Flexfield ranges assigned to your budget organization, your Oracle General Ledger application dynamically creates a new Accounting Flexfield if budgeting is allowed but does not assign it to your budget organization.

You can upload budgets for multiple sets of books at the same time. And, each of these sets of books can have a different Accounting Flexfield structure. You indicate which set of books your budget amounts are for by entering the identification number for the set of books in the Set_of_Books_ID column of the Budget Interface table. To determine the identification number of your sets of books, you need to reference the underlying database table GL_SETS_OF_BOOKS.

You then enter each Accounting Flexfield by entering valid segment values for each segment of the Accounting Flexfield. You need to enter a segment value for each segment you enabled in your Oracle General Ledger application. Budget Upload assumes you enter values for Accounting Flexfield segments into your spreadsheet in the same order as you store them in your Oracle General Ledger application. To determine the order in which your Oracle General Ledger application stores your Accounting Flexfield segments, you need to reference the underlying database table GL_CODE_COMBINATIONS.

For each set of books and associated Accounting Flexfield structure, the Code Combinations table stores every Accounting Flexfield you use in your Oracle General Ledger application. The Code Combinations table maintains these Accounting Flexfields by storing segment value information in columns Segment1 through Segment30 in the table. For instance, if you have two sets of books where the first set of books uses a six-segment Accounting Flexfield structure and the second set of books uses a five-segment Accounting Flexfield structure, your Oracle General Ledger application maintains Accounting Flexfield information for the first structure using six segment columns in the Code Combinations table and maintains Accounting Flexfield information for the second structure using five segment columns.

Accounting Flexfield Structures Created Before Release 10

Before Release 10, when you set up a new Accounting Flexfield structure and begin defining new Accounting Flexfields, your Oracle General Ledger application randomly chooses segment columns in the Code Combinations table in which to store your Accounting Flexfields. Thus, your Oracle General Ledger application may or may not use a continuous range of columns for each of your Accounting Flexfield structures. For example, when you set up your six-segment Accounting Flexfield structure for your first set of books, your Oracle General Ledger application might have chosen columns Segment1 through Segment6 in which to store your Accounting Flexfields. Similarly, when you set up your five-segment Accounting Flexfield structure for your second set of books, your Oracle General Ledger application might have chosen columns Segment3 through Segment6 and Segment 11 in which to store your Accounting Flexfields.

Therefore, before you enter Accounting Flexfield segment values into the segment columns of your spreadsheet and Budget Interface table, you need to review the Code Combinations table to determine which set of columns your Oracle General Ledger application uses for each Accounting Flexfield structure. For example, if you have a six-segment structure where your first segment is the balancing segment, your second segment is department, your third segment is account, and so on, you need to check the Code Combinations table to determine in which column your Oracle General Ledger application stores your balancing segment values, which column your Oracle General Ledger application stores your department values, and so on. You then need to assign values to the same segment columns in your spreadsheet. Contact your system administrator or MIS specialist for assistance in viewing the Code Combinations table.

Accounting Flexfield Structures Created in Release 10

In Release 10, when you define a new Accounting Flexfield structure, you choose which columns in the Code Combinations table you want your Oracle General Ledger application to store your Accounting Flexfield segment values. Before you enter Accounting Flexfield segment values into the segment columns of your spreadsheet and Budget Interface table, determine to which column you assigned each segment using the Define Key Flexfield Segments form.

Make sure you specify segment values correctly. For example, value '01' is not the same as value '1'. Display size tells you how wide Budget Upload expects each segment value to be. For example, if display size is three, then your segment value would be '100,' but if

your display size is four, then your segment value would be '0100'. You can determine the correct display size and attributes (alphabetic, numeric, right-justify zero-fill, and so on) for each segment in your Accounting Flexfield using the Define Value Set form.



Define Key Flexfield Segments

Define Value Sets

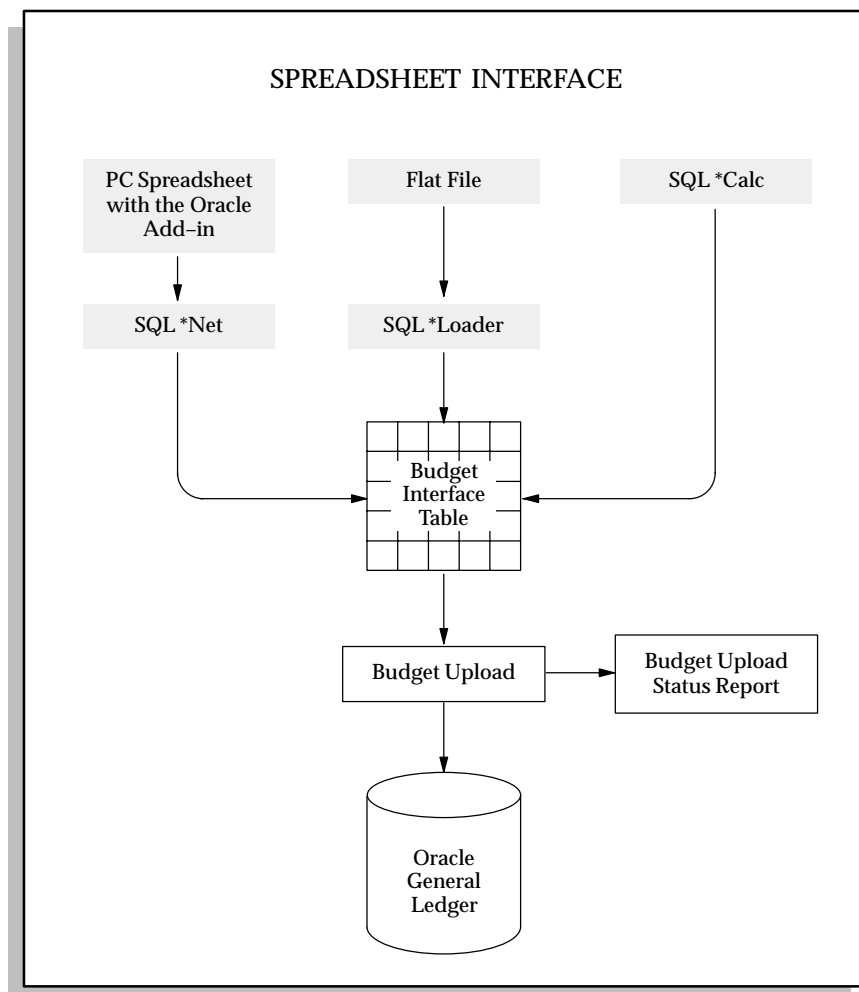
(Oracle Applications Flexfields Manual)

Uploading Your Budget Information

Before you run Budget Upload, you must first transfer your spreadsheet to your server. The procedure to follow depends on the software you use.

Optionally, you can use a spreadsheet application to create your budget, then use Oracle Glue or Dynamic Data Exchange (DDE) to populate GL_BUDGET_INTERFACE.

Figure 4 – 3 Spreadsheet Interface



Software Requirements

Choose a spreadsheet that provides the most flexibility for your organization. The only software requirement is that you choose a spreadsheet application that can store data in a ASCII format. You will also need a file transfer program to transfer the flat file to your host computer.

Transfer your Spreadsheet to your Host Computer

Before you run Budget Upload, you must first transfer your spreadsheet to your host computer. The procedure to follow depends on the software you use.

For example, if you work with Microsoft Excel, use / **Print File** to create an ASCII file from your spreadsheet. Specify the range of data you want to transfer to the Budget Interface Table. Use the following margins:

Left Margin	=	0
Right Margin	=	240
Top Margin	=	0
Bottom Margin	=	0

Use / **Print Other** to specify the file type for the ASCII file as unformatted. Use / **Print Go** to actually produce the ASCII file on your personal computer. Your ASCII file will have .PRN as its extension.

Next, transfer the ASCII file to your host computer using any file transfer software.

Load the Budget Interface Table

Use SQL*Loader to move information from file to GL_BUDGET_INTERFACE table. SQL*Loader is provided as part of the Oracle RDBMS. You need to create a control file (e.g. budget.ctl) and use it with SQL*Loader to load the data from your .PRN file to the Budget Interface Table. Use the following command to run SQL*Loader:

```
sqlload <username/password> <control_file>.ctl <log_file>.log
```

where <username/password> are the username and password of the database where you are loading the data, <control_file>.ctl is the control file you created (e.g. budget.ctl) and <log_file>.ctl is a log file that contains any messages of the SQL*Loader process.

Once you have successfully loaded the Budget Interface table, use Budget Upload to update budget balances.



Upload Budgets

(Oracle General Ledger Reference Manual or Oracle Government General Ledger Reference Manual)

Running Budget Upload

Running Budget Upload can be a one-step or multi-step process depending on the contents of your spreadsheet. If your spreadsheet addresses all required columns in the Budget Interface table, then you can successfully run it in one step. However, if you do not address all required columns in the Budget Interface table, or if you enter data into your spreadsheet that is not valid in your Oracle General Ledger application, then your Oracle General Ledger application informs you of errors in the Budget Spreadsheet Upload Status Report. In this case you need to correct the errors and rerun Budget Upload.

Run Budget Upload using the Upload Budgets form. Enter the name of the budget you want your Oracle General Ledger application to upload. Your Oracle General Ledger application retrieves budget information for this budget only. Also enter the name of the budget organization you want your Oracle General Ledger application to upload. Your Oracle General Ledger application retrieves budget information for this budget organization only and updates your budget account balances. You can also upload budget information for all of your budget organizations by entering the 'All' budget organization.



Define Budget Organization
(Oracle Government General Ledger Reference Manual)

Budget Validation

Budget Upload validates all of your budget information for compatibility with your Oracle General Ledger application. Budget Upload validates your data by ensuring that the columns of the Budget Interface table reference the appropriate values and columns in your spreadsheet. Budget Upload also checks to make sure that the following conditions are true:

- Your account is assigned to a budget organization
- The budget entry type for your account is Entered
- Your budget is not Frozen
- Your budget organization is not Frozen
- Your budget fiscal year is open for your budget

Reviewing the Budget Spreadsheet Upload Status Report

Use the Budget Spreadsheet Upload Status report to review the status of all budget information you transfer from your spreadsheet to your Oracle General Ledger application. The Budget Spreadsheet Upload

Status report prints a separate line for each budget/budget organization combination for which you uploaded budget information. The report lists the number of posted accounts and the number of unposted accounts for the combination.

Your Oracle General Ledger application automatically deletes budget records in the Budget Interface table for all accounts it can successfully update. Your Oracle General Ledger application does not delete budget records in the Budget Interface table or update budget balances for accounts containing errors. If Budget Upload runs across any problems, such as invalid data during the upload process, it lists the accounts containing errors as well as the associated budget and budget organization in the Budget Spreadsheet Upload Status report. Once you correct any errors, rerun Budget Upload.



Budget Spreadsheet Upload Status Report

(Oracle General Ledger Reference Manual or Oracle Government General Ledger Reference Manual)

Correcting Errors

Correct errors based on the error code you receive on the Budget Spreadsheet Upload Status report. You may need to make changes in your Oracle General Ledger application or in your spreadsheet. For example, if you receive an error code indicating that an account is unposted because the budget for this account has a status of Frozen, you can change the status of the budget in your Oracle General Ledger application. If you receive an error code indicating that an account is unposted because you specified an incorrect update logic type, you can enter the correct update logic type in the Update_Logic_Type column of your spreadsheet.

The following errors may occur if you violate budget upload validation rules:

- This account has not been assigned to any budgeting organization.
- The budget is 'Frozen or 'Inactive'.
- The organization is 'Frozen' for this budget.
- The organization is 'Inactive'.
- The account is not assigned to the given organization.
- The account is not assigned to the given organization as 'Entered'.
- The account has an invalid currency code.

- The account was not specified as Add ('A') or Replace ('R').
- The account appeared more than once in a specified budget and organization.
- The organization specified does not exist.
- The fiscal year specified for the budget is not open or does not exist.

Reporting on your Budgets

There are several ways you can review and report budget information. You can perform an online inquiry to get the information you need right away. Simply specify the period range, currency, factor, budget name, amount types and the Accounting Flexfield or range of Accounting Flexfields in the Account Inquiry form and your Oracle General Ledger application displays complete actual, budget and variance amounts online.

You can also request a variety of budget variance reports which you design using the Financial Statement Generator. These reports can include budget, actual, variance and variance percentage amounts.



Overview of Financial Reporting
(*Oracle General Ledger Reference Manual* or *Oracle Government General Ledger Reference Manual*)



Account Inquiry
(*Oracle General Ledger Reference Manual* or *Oracle Government General Ledger Reference Manual*)

Oracle Inventory Open Interfaces

This chapter contains information about the following Oracle Inventory open interfaces:

- Open Demand Interface
- Open Transaction Interface
- Open Replenishment Interface
- Open Item Interface

Open Demand Interface

The Open Demand Interface provides all the functions you need to interface an external order entry system with Oracle Inventory and Oracle Manufacturing applications. It provides a two-way interface that lets you:

- Provide visibility to demand created in external applications for forecasting, planning, and order promising purposes. This includes the option to automatically check ATP when adding demand to verify availability.
- Reserve on-hand inventory to specific sources of demand (for example, sales orders). This includes the option to 'auto-reserve' inventory at a detailed level for warehouse picking.
- Check Available to Promise (ATP), Available to Reserve (ATR), and on-hand quantity information.
- Specify Assemble to Order Model and Option demand details for creation of and/or association with an ATO Configured Item.

The purpose of this essay is to explain how to use the Open Demand Interface to integrate other applications with Oracle Inventory.

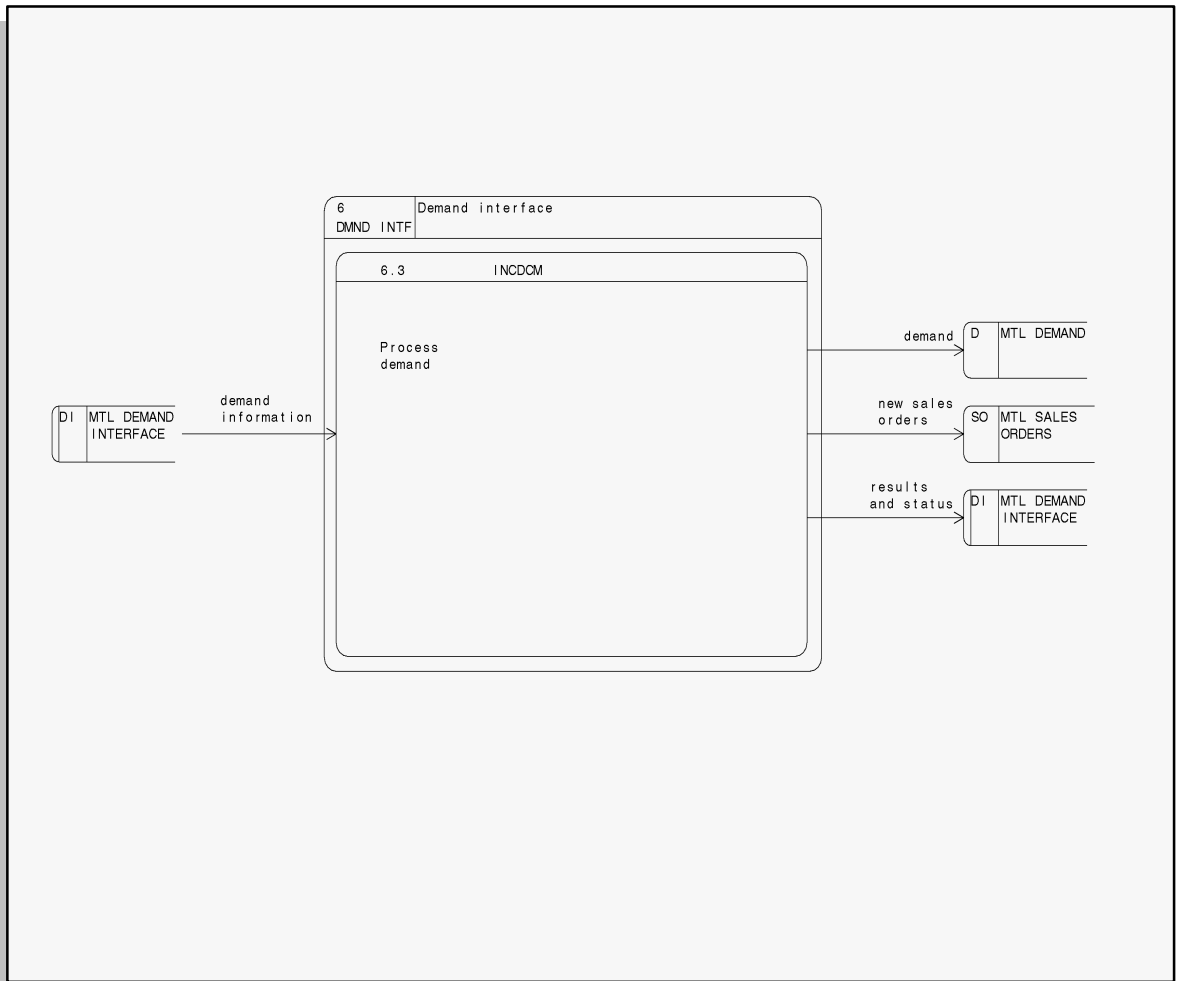
Functional Overview

The Demand Interface supports several functions or 'actions' that you can use to perform the above operations. These include:

- Demand Add
- Reservation Add
- Reservation Modify
- Demand and Reservation Add
- Update Forecast Attributes
- Pick Release
- ATS Query (request available and on-hand quantity information)
- ATS QuickPick (request available and on-hand quantity details)
- ATP Check

The following data flow diagram shows the key tables and programs that comprise the Open Demand Interface:

Figure 5 – 1



To use the Open Demand Interface, you need to insert rows into `MTL_DEMAND_INTERFACE` that pass all required information to execute the above functions. The rows you insert contain action codes (`ACTION_CODE`) that tell the Demand Manager which function(s) to execute. You group your interface transactions using a common identifier `SCHEDULE_GROUP_ID`. If any row in the group fails

processing, the Demand Manager fails all rows. The Demand Manager only processes rows with `PROCESS_FLAG = 1` (Yes) and `ERROR_CODE = null`.

When you are placing or updating demand and reservations, the flow of data passes through the interface table to `MTL_DEMAND`. In some cases, you pass a unique identifier that you can subsequently use to query results from `MTL_DEMAND` on rows that were created or modified by your request. If errors occur, the Demand Manager leaves the inserted row in the interface table and updates it with an error code and explanation.

When you are using the Open Demand Interface to query for on-hand, available to reserve, or available to promise information, the Demand Manager takes the rows inserted into the interface table as input, and either returns the results by updating the same row, or by inserting new rows back into the interface table with a unique group identifier for querying purposes.

Transaction Processing Modes

There are two basic ways that you can interface with the Demand Manager. The first way is to simply commit rows to the interface table, leaving them to be picked up and processed by the polling Demand Manager. This mode is known as 'Background' processing.

The second way is to insert the rows into the interface table and call the Demand Manager Processor directly from a form (via user-exit) or concurrent program. This gives more direct feedback to the calling function, providing a more timely, integrated solution.

You may choose to use a combination of both methods, using the background mode (`TRANSACTION_MODE = 3`) to maintain demand information, and using on-line or concurrent mode (`TRANSACTION_MODE= 2, 1`) to check available and on-hand information. See the next section for details on how to call the Demand Processor from a form or C program.

Setting Up the Demand Interface

Setting Up the Inventory Concurrent Manager

For optimal processing in the Demand Interface, you need to set up your concurrent manager to best handle your transaction volumes while balancing your performance requirements and your system load

restrictions. Oracle Inventory ships the Demand Manager to be run in Inventory's own concurrent manager named Inventory Manager. It is defaulted to run in the Standard workshift with Target Processes = 1 and Sleep Time of 60 seconds.

With this configuration, the Demand Manager must contend with other processes such as the Material Transaction Manager and Transaction Workers for the same processing queue. If you have the available resources, you can substantially reduce the time to process your interfaced demand requests by increasing the target processes and reducing the concurrent manager sleep time using the Define Concurrent Managers form. This will allow the Demand Manager to run in parallel with the Transaction Manager and Transaction Workers.

Starting the Demand Manager

Once you have set up the Inventory concurrent manager, you can launch the Inventory Demand Manager in Oracle Inventory's Request Interface Managers form. This launches the Demand Manager and lets you specify the interval to use to poll the interface table.

Calling the Demand Processor from a Form

To call the Demand Processor from a form, you need to first populate MTL_DEMAND_INTERFACE, and then call the Oracle Inventory user-exit GROUP_PROCESS with tokens of GROUP_ID and PARTIAL_FLAG as follows:

```
# INV GROUP_PROCESS
  GROUP_ID=group_id
  [PARTIAL_FLAG=partial_flag]
```

group_id is required and identifies which rows in MTL_DEMAND_INTERFACE to process (using SCHEDULE_GROUP_ID).

partial_flag defaults to 2 (No), but can be set to 1 (Yes) to indicate that demand rows in the group can succeed even if others fail.

Calling the Demand Processor from a C Program

To call the Demand Processor from a C program, you need to first populate MTL_DEMAND_INTERFACE and then call the GrpDemIface function with the following function prototype. This function is archived in the Oracle Inventory object library (libinv.a), which references functions defined in (libfnd.a):

```
sb4 GrpDemIface(/*_ sb4 GroupId, sb2 ProcessMode,  
                sb2 PAFlag _*/);
```

sb2 and sb4 are type definitions Signed Short and Signed Long.

GroupId identifies which rows in MTL_DEMAND_INTERFACE to process (using SCHEDULE_GROUP_ID).

ProcessMode should match the value in MTL_DEMAND_INTERFACE.TRANSACTION_MODE for the rows you are processing. You should use either mode 1 or 2. With mode 2, output is directed to a log file. This should only be used for concurrent processing. Mode 1 does not write to a log file, so you should use it when calling the Demand Processor from a user-exit.

PAFlag identifies whether partial completions of the group request are allowed. Set this flag to 1 (Yes) to indicate that demand rows in the group can succeed even if others fail, 2 (No) to indicate that all must complete successfully as a group.

Setting Up Your Sales Order Flexfield

Oracle Inventory uses a flexfield to hold the unique Sales Order name so that it does not need to join back to the feeder Order Entry system. This means that you must set up Inventory's Sales Order flexfield (MKTS) using the Define Key Flexfield Segments form with enough segments so that the combination is unique across all orders.

For example, Oracle Order Entry guarantees uniqueness within an installation, order type, and order number. Consequently, standard installation steps require that you set up a three segment. If you can guarantee that one segment is sufficient (for example, Order Number), then that is all you need to enable in your flexfield definition.

When you enter rows into the demand interface for sales order demand, you should use the Sales Order segment values to identify the order. The Demand Manager will validate against MTL_SALES_ORDERS, and if the code combination does not already exist will create a new one. All references to the order number internal to Inventory in reports and inquiries will be based on this relationship.

Demand Interface Function Descriptions

This section describes each action that you can perform using the Demand Interface. It also includes for each action a list of the required and optional inputs. Since several of the demand actions provide a

corresponding output through the MTL_DEMAND_INTERFACE table, the columns where the output can be found are also listed.

All functions that you perform on the demand table require that you specify a 'demand key' which contains the complete set of information to uniquely identify rows in MTL_DEMAND. The demand key contains the following:

- Demand source type (for example, Sales Order, Account)
- Demand source. The demand table provides several columns for you to uniquely identify your demand source, such as the order, line, and shipment details. The columns are:
 - DEMAND_SOURCE_HEADER_ID
or
 - DEMAND_HEADER_SEGMENT1 – 30 (for sales order, account, or account alias source types)
 - DEMAND_SOURCE_NAME (for user-defined transaction source types)
 - DEMAND_SOURCE_LINE
 - DEMAND_SOURCE_DELIVERY
- Organization
- Item

Demand Add

Demand Add lets you place new demand and update existing demand for a given demand source in the demand table. In addition to the demand key, you must specify the demand quantity in any valid unit of measure for the item using LINE_ITEM_QUANTITY and LINE_ITEM_UOM. The Demand Manager will automatically convert the quantity into a quantity in the item's primary unit of measure. You must also specify the REQUIREMENT_DATE indicating when you expect the demand to be fulfilled. In an ATO scenario, you must identify whether the DEMAND_TYPE is:

1. Models
2. Option Classes
3. Options

Otherwise, for standard demand, you need to specify DEMAND_TYPE of 6 (Standard).

If the demand you are creating is specific to a subinventory, you should enter the value in SUBINVENTORY. Note that this automatically restricts any subsequent reservations for this demand to available quantities within the specified subinventory.

MRP Forecast Attributes

You *must* specify the following demand attributes used by MRP for forecast consumption and loading the master demand schedule:

- CUSTOMER_ID
- BILL_TO_SITE_USE_ID
- SHIP_TO_SITE_USE_ID

You can optionally specify the following demand attribute:

- TERRITORY_ID

In an ATO Scenario, you can update these forecast attributes for all demand under a model in a single call by entering a value in RTO_MODEL_SOURCE_LINE corresponding to the value in DEMAND_SOURCE_LINE for the parent Model.

You can also enter and update the Demand Class as another option to partition your demand. This information is used for forecast consumption and/or to check ATP if you are using an ATP Rule that partitions supply and demand by Demand Class.

ATP Attributes

When you interface demand, you can request that ATP be checked as part of the validation process before the demand is placed by setting ATP_CHECK to 1 (Yes). If you want to place demand for several demand rows only if all rows pass the ATP check, then you should also populate the ATP_GROUP_ID with a common identifier for all rows in the group.

If the ATP checks succeed, demand will be updated and the interfaced rows will be removed from the MTL_DEMAND_INTERFACE table.

Please see the explanation of the ATP Check function for information on populating ATP attributes.

ATO/BOM Attributes

When you place demand for ATO models, options, and option classes, the Demand Manager automatically identifies them from item master

attributes. In certain scenarios, the Demand Manager automatically explodes down the bill of materials to create interface rows for ATP-able components (see ATP Check).

Once the ATP check succeeds (if required) the Demand Manager explodes the model, option, and option class bills of material to populate MTL_DEMAND with demand rows for any ATP-able components. You need to populate the following columns:

- **COMPONENT_SEQUENCE_ID.** Unique identifier in BOM structure. For Model demand, this should correspond to BOM_BILL_OF_MATERIALS.BILL_SEQUENCE_ID. For Option or Option Class demand, this should correspond to BOM_INVENTORY_COMPONENTS.COMPONENT_SEQUENCE_ID.
- **PARENT_COMPONENT_SEQ_ID.** BOM structure unique identifier of parent.
- **RTO_MODEL_SOURCE_LINE.** The Model's DEMAND_SOURCE_LINE for all Options, Option Classes, and Components.
- **EXPLOSION_EFFECTIVITY_DATE.** Date used to explode bill of materials
- **CONFIG_STATUS.** Status code to indicate whether the model, option, option class demand rows are complete (in other words, ready for configuration item creation). The values are:
 - 10 – Not ready for configuration item creation
 - 20 – Ready for configuration item creation

Reservation Add

Reservation Add lets you reserve available inventory for demand that has already been placed. You need to specify the demand key, as well as the new reservation balance and UOM in LINE_ITEM_RESERVATION_QTY, LINE_ITEM_UOM. Reservation Add will determine if a reservation already exists with the exact same details, and either replace the existing reservation quantity, or create a new reservation row if none already exists. The following columns can optionally be used to reserve inventory to greater detail than organization and item.

- REVISION
- LOT

- SUBINVENTORY
- LOCATOR_ID or LOC_SEGMENT1 – 20

You can only reserve revision, lot, and locator details if that quantity control level is enabled. Also, if you specify reservation details, you must make sure to specify any other details higher up in the reservation hierarchy (see above list). For example, if an item is under lot control, but not revision control, and you want to reserve material in a specific subinventory, you must specify the organization, item, lot, and subinventory (no revision).

You can have multiple reservations for each demand row, but the total reserved quantity cannot exceed the demand.

Reservation Modify

Reservation Modify performs much like Reservation Add, except you need to specify the net change from the current reservation quantity in `LINE_ITEM_RESERVATION_QTY`. If no reservation exists for the same organization, item, revision, subinventory, and lot, Reservation Modify will create a new row. This function requires that you specify `RESERVATION_TYPE=2` for reservations of available inventory.

See Reservation Add for more details.

Demand and Reservation Add

This function combines the capabilities of Demand Add and Reservation Add, letting you create demand and reserve all or part of the demand quantity in one call. If the demand row does not already exist, then a new one will be created. If the demand and/or reservation rows already exist, they will be updated to the new state.

See Demand Add and Reservation Add for more details.

Update Forecast Attributes

This function can be used to update attributes used by MRP to consume the item forecast. The attributes are `CUSTOMER_ID`, `TERRITORY_ID`, `BILL_TO_SITE_USE_ID`, and `SHIP_TO_SITE_USE_ID` (see Demand Add for details). In an ATO scenario, you automatically update the attributes for exploded demand rows under a parent model, option, or option class. You can also update the attributes for all demand under a model in one call by entering a value in

RTO_MODEL_SOURCE_LINE corresponding to the value in DEMAND_SOURCE_LINE for the parent Model.

Pick Release

This function takes existing demand and attempts to reserve inventory to the item's lowest inventory control level. It uses the Picking Rule defined for an item to determine which order to search for available inventory to reserve. Any existing reservations that do not completely specify all control options are replaced with 'detailed' reservations.



Pick Release
Scheduling Orders
(Oracle Order Entry Reference Manual)

You need to specify what you want pick released using the demand key, and how much quantity to reserve using LINE_ITEM_QUANTITY and LINE_ITEM_UOM. You can set the PARTIALS_ALLOWED_FLAG to 1 (Yes) to indicate that Pick Release should reserve as much as possible even if the total cannot be satisfied. If this flag is not set and the entire quantity cannot be reserved, the request fails.

You need to also specify a unique identifier in AUTODETAIL_GROUP_ID. Pick Release passes this value through to MTL_DEMAND so you can query up reservation details created by your request.

If you set SINGLE_LOT_FLAG to 1 (Yes), your request only will only succeed if the entire requested quantity can be fulfilled with a single lot. Pick Release searches for a complete lot quantity using the Picking Rule criteria. For items under lot expiration control, you can optionally populate LOT_EXPIRATION_CUTOFF_DATE to eliminate lots from consideration whose expiration date is earlier from inclusion in the results.

Pick Release returns the following results in the interface table:

- AUTODETAILED_QUANTITY. The total quantity successfully reserved. The column is left null if partials are not allowed and the full quantity was not able to be reserved.
- AVAILABLE_TO_RESERVE – The total quantity available to reserve regardless of whether the request succeeded
- QUANTITY_ON_HAND

ATS Query (Single-Row Availability Check)

You can use this function to request information about the availability of inventory to reserve or transact. You must specify the demand key, organization, item, and any optional item control details for which you want information. The Demand Manager returns the requested information in QUANTITY_ON_HAND and AVAILABLE_TO_RESERVE.

For items under lot expiration control, you can optionally populate LOT_EXPIRATION_CUTOFF_DATE to eliminate lots from consideration whose expiration date is earlier from inclusion in the results.

ATS Quick Pick (Multi-Row Detail Availability Check)

This function lets you request availability information much like ATS Query. The primary difference is that you specify at what control level you want the available quantity results to be returned. For example, you can request the availability of all lots with on-hand quantity within a given organization. To accomplish this, you need to pass the organization and item in question, and specify a control level of 'Lot' in RESERVE_LEVEL. The valid control levels are:

2 – Revision

3 – Lot

5 – Subinventory

6 – Locator

Instead of updating the original interface row, new rows are inserted into MTL_DEMAND_INTERFACE with the same SCHEDULE_GROUP_ID from the 'input row'. The results are populated in QUANTITY_ON_HAND and AVAILABLE_TO_RESERVE along with the corresponding control details (for example, revision, lot).

As with ATS Query, for items under lot expiration control, you can optionally populate LOT_EXPIRATION_CUTOFF_DATE to eliminate lots from consideration whose expiration date is earlier from inclusion in the results.

ATP Check

This function lets you check ATP for a given item within an organization. You must define the whole demand key, as well as the

requested quantity in `LINE_ITEM_QUANTITY` (if you just want the current ATP information, you can specify a requested quantity of zero). All other inputs listed below are optional.

- `LINE_ITEM_QUANTITY`, `LINE_ITEM_UOM`. The quantity requested. If quantity entered is greater than zero, ATP Check will return the earliest date this quantity is available. If the unit of measure is not specified, the item's primary uom is used as a default.
- `REQUIREMENT_DATE`. The date for which the available quantity is requested. If entered, ATP Check will return the available to promise quantity on that date.
- `ATP_RULE_ID`. If you populate this column, ATP Check will use it to override the default ATP rule on the Item Master. It must be a valid foreign key reference to a rule defined in `MTL_ATP_RULES`.
- `ATP_CALENDAR_ORGANIZATION_ID`. If specified, ATP Check uses the workday calendar for this organization. Otherwise, the workday calendar for the organization against which the ATP check is being performed is used.
- `ATP_CHECK`. This column overrides the Item Master default. Set it to 1 to enforce an ATP check, 2 to enforce that no ATP check is done. This override should be used carefully since inconsistent use when placing demand could invalidate demand that was previously 'allocated' using ATP Check.
- `ATP_COMPONENTS_FLAG`. This column overrides the Item Master default. Set it to 1 to force explosion of an item's bill of materials to include ATP-able components. If you leave this flag null, ATP Check will still explode to find ATP-able components if `MTL_SYSTEM_ITEMS.ATP_COMPONENTS_FLAG = 1` (Yes) and `BOM_INVENTORY_COMPONENTS.CHECK_ATP = 1` (Yes). As with `ATP_CHECK`, you should be careful in your use of this flag since inconsistent use while placing demand could invalidate demand that was previously 'allocated' using ATP Check.
- `LATEST_ACCEPTABLE_DATE`. This limits the date ATP Check will return as the `GROUP_ATP_DATE`.
- `DEMAND_CLASS`. If the ATP Rule in use is defined for Demand Class ATP, this column identifies the subset of supply and demand on which to base the query.

- **ATP_LEAD_TIME.** The number of days earlier than the ATP date that the item must be available. In an ATO scenario, you need to specify the assembly lead times for option classes and options to get an accurate representation of when the subassemblies and components will be required. This is also populated by the demand manager when it explodes to find demand for ATP-able components.

ATP Check returns multiple results by updating the same interface row. The output columns are:

- **REQUEST_DATE_ATP_QUANTITY.** The available quantity on the request date regardless of whether the quantity is satisfied.
- **EARLIEST_ATP_DATE.** The earliest date that can satisfy the requested quantity regardless of the request date.
- **EARLIEST_ATP_DATE_QUANTITY.** The maximum quantity available on the earliest ATP date.
- **REQUEST_ATP_DATE.** The first date on or after the required date where the available quantity is enough to satisfy the requested quantity.
- **REQUEST_ATP_DATE_QUANTITY.** The maximum available quantity on the requested ATP date.
- **INFINITE_TIME_FENCE_DATE.** The date offset from the current date by the Infinite Supply Time Fence.

In addition to checking ATP for items one at a time, you can use the Demand Interface to perform a 'Group ATP' check. You pass multiple rows containing different item/organization combinations, and populate **ATP_GROUP_ID** with a common group identifier. ATP Check not only returns all information for each row as described above, but it also determines the earliest date, if any, before the requirement date that all requests can be satisfied. This result is returned in **GROUP_AVAILABLE_DATE**.

Inserting into the Demand Interface Table

MTL_DEMAND_INTERFACE Table Description – Inputs

The following graphic describes the input columns to the **MTL_DEMAND_INTERFACE** table cross-referenced against the Open Demand Interface functions. Required columns are identified with an 'R', optional columns with an 'O', and derived columns with a 'D'.

MTL_DEMAND_INTERFACE Column Name	Type	Demand Add	Reserva- tion Add	Reserva- tion Modify	Demand /Reserve Add	Update Attr	Pick Release	ATS Query	ATS Quick Pick	ATP Check
SCHEDULE_GROUP_ID	Number	R	R	R	R	R	R	R	R	R
TRANSACTION_PROCESS_ORDER	Number	O	O	O	O	O	O	O	O	O
DEMAND_SOURCE_TYPE	Number	R	R	R	R	R	R	R	R	R
DEMAND_SOURCE_HEADER_ID or DEMAND_HEADER_SEGMENT1 - 30 or DEMAND_SOURCE_NAME	Varchar2(40)	R	R	R	R	R	R	R	R	R
DEMAND_SOURCE_LINE	Varchar2(30)	R	R	R	R	R	R	R	R	R
DEMAND_SOURCE_DELIVERY	Varchar2(30)	R	R	R	R	R	R	R	R	R
USER_LINE_NUM	Varchar2(30)	O			O					
USER_DELIVERY	Varchar2(30)	O			O					
EXTERNAL_SOURCE_CODE	Varchar2(30)	O	O	O	O	O	O	O	O	O
EXTERNAL_SOURCE_LINE_ID	Number	O	O	O	O	O	O	O	O	O
ACTION_CODE	Number	R	R	R	R	R	R	R	R	R
PROCESS_FLAG	Number	O	O	O	O	O	O	O	O	O
LOCK_FLAG	Number									
VALIDATE_ROWS	Number	O	O	O	O	O	O	O	O	O
TRANSACTION_MODE	Number	R	R	R	R	R	R	R	R	R
ORGANIZATION_ID or ORGANIZATION_NAME	Number Varchar2(60)	R	R	R	R	R	R	R	R	R
INVENTORY_ITEM_ID or ITEM_SEGMENT1 - 20	Number Varchar2(40)	R	R	R	R	R	R	R	R	R
REVISION	Varchar2(3)		O	O	O			O	O	
LOT_NUMBER	Varchar2(30)		O	O	O			O	O	
SUBINVENTORY	Varchar2(10)	O	O	O	O			O	O	
LOCATOR_ID or LOC_SEGMENT1 - 20	Number Varchar2(40)		O	O	O			O	O	
LINE_ITEM_QUANTITY	Number	R			R		R			
LINE_ITEM_RESERVATION_QTY	Number		R	R	R		R			
LINE_ITEM_UNIT_OF_MEASURE or LINE_ITEM_UOM	Varchar2(25) Varchar2(3)	O	O	O	O		O	O	O	O

Table 5 – 1 Open Demand Interface Table – Input Columns (Page 1 of 3)

MTL_DEMAND_INTERFACE Column Name	Type	Demand Add	Reserva- tion Add	Reserva- tion Modify	Demand /Reserve Add	Update Attr	Pick Release	ATS Query	ATS Quick Pick	ATP Check
PRIMARY_UOM_QUANTITY	Number	D			D		D	D	D	
PRIMARY_UOM	Varchar2(3)	D	D	D	D		D	D	D	
RESERVATION_QUANTITY	Number		D	D	D					
REQUIREMENT_DATE	Date	0			0					0
DEMAND_TYPE	Number	R			R					
RESERVATION_TYPE	Number			R						
AUTODETAIL_GROUP_ID	Number						R			
SINGLE_LOT_FLAG	Number						O			
RESERVE_LEVEL	Number						O			
DEMAND_CLASS	Varchar2(30)	O			O					O
CUSTOMER_ID	Number	O			O	R				
TERRITORY_ID	Number	O			O	R				
BILL_TO_SITE_USE_ID	Number	O			O	R				
SHIP_TO_SITE_USE_ID	Number	O			O	R				
LOT_EXPIRATION_CUTOFF_DATE	Date						O	O	O	
PARTIALS_ALLOWED_FLAG	Number						O			
ATP_CHECK	Number	O			O					O
ATP_GROUP_ID	Number	O			O					O
ATP_RULE_ID	Number	O			O					O
ATP_COMPONENTS_FLAG	Number	O			O					O
LATEST_ACCEPTABLE_DATE	Date	O			O					O
ATP_CALENDAR_ORGANIZATION_ID	Number	O			O					O
ATP_LEAD_TIME	Number	O			O					O
BOM_LEVEL	Number									
EXPLOSION_GROUP_ID	Number									
EXPLOSION_EFFECTIVITY_DATE	Date	O			O					
COMPONENT_SEQUENCE_ID	Number	O			O					
PARENT_COMPONENT_SEQ_ID	Number	O			O					
CONFIG_STATUS	Number	O			O					

Table 5 – 1 Open Demand Interface Table – Input Columns (Page 2 of 3)

MTL_DEMAND_INTERFACE Column Name	Type	Demand Add	Reserva- tion Add	Reserva- tion Modify	Demand /Reserve Add	Update Attr	Pick Release	ATS Query	ATS Quick Pick	ATP Check
RTO_MODEL_SOURCE_LINE	Varchar2(30)	O			O	O				
ATTRIBUTE_CATEGORY	Varchar2(30)	O			O					
ATTRIBUTE1 – 15	Varchar2(240)	O			O					

Table 5 – 1 Open Demand Interface Table – Input Columns (Page 3 of 3)

MTL_DEMAND_INTERFACE Table Description – Outputs

The following graphic describes the output columns from the MTL_DEMAND_INTERFACE table cross-referenced against the Open Demand Interface functions. Several functions by their nature are only entering/updating information in MTL_DEMAND, and therefore will only get an 'output' if an error occurs. Others, such as ATP Check, expect outputs as a result of successful completion.

MTL_DEMAND_INTERFACE Column Name	Type	Demand Add	Reserva- tion Add	Reserva- tion Modify	Demand /Reserve Add	Update Attr	Pick Release	ATS Query	ATS Quick Pick	ATP Check
ERROR_CODE	Number	✓	✓	✓	✓	✓	✓	✓	✓	✓
INFINITE_TIME_FENCE_DATE	Date	✓			✓					✓
REQUEST_DATE_ATP_QUANTITY	Number	✓			✓					✓
REQUEST_ATP_DATE	Date	✓			✓					✓
REQUEST_ATP_DATE_QUANTITY	Number	✓			✓					✓
EARLIEST_ATP_DATE	Date	✓			✓					✓
EARLIEST_ATP_DATE_QUANTITY	Number	✓			✓					✓
GROUP_AVAILABLE_DATE	Date	✓			✓					✓
AUTODETAILED_QUANTITY	Number						✓			
AVAILABLE_TO_RESERVE	Number						✓	✓	✓	
QUANTITY_ON_HAND	Number						✓	✓	✓	

Table 5 – 2 Open Demand Interface Table – Output Columns (Page 1 of 1)

DEMAND_SOURCE_TYPE

This column identifies the origination of the demand or reservation. The valid values for pre-defined source types are:

2 – Sales Order

3 – Account

6 – Account Alias

User-defined source types are defined in MTL_TXN_SOURCE_TYPES. This column is a foreign key reference to TRANSACTION_SOURCE_TYPE_ID.

DEMAND_SOURCE_HEADER_ID

DEMAND_SOURCE_HEADER_ID or the corresponding flexfield segment columns (DEMAND_HEADER_SEGMENT1 –30) are required for all transaction source types other than those that are user-defined. You should enter the foreign key ID that points to the context table identified by the DEMAND_SOURCE_TYPE.

Source Type	Foreign Key Reference
Account	GL_CODE_COMBINATIONS.CODE_COMBINATION_ID
Account Alias	MTL_GENERIC_DISPOSITIONS.DISPOSITION_ID
Sales Order	MTL_SALES_ORDERS.SALES_ORDER_ID

**Table 5 – 3 DEMAND_SOURCE_HEADER_ID,
Foreign Key References (Page 1 of 1)**

DEMAND_HEADER_SEGMENT1 – 30

You can use these flexfield segment columns instead of DEMAND_SOURCE_HEADER_ID for pre-defined source types (in other words, Account, Account Alias, or Sales Order). For example, if you were creating a reservation to an account, you would enter the segment values of the GL Code Combination in these columns instead of using the internal identifier.

DEMAND_SOURCE_NAME

This column is required for user-defined source types. Enter the value of the source name, such as an order number, to be displayed on all reports and inquiries related to demand.

DEMAND_SOURCE_LINE

This column is one of three 'levels' of the demand key that help you uniquely identify your demand row. The Demand Interface takes all three levels (header, line, delivery) as a whole, and does not generally distinguish between them.

DEMAND_SOURCE_DELIVERY

This column is one of three 'levels' of the demand key that help you uniquely identify your demand row. The Demand Interface takes all three levels (header, line, delivery) as a whole, and does generally distinguish between them.

USER_LINE_NUM

This column is an optional field that can be used to identify specific demand details in WIP and MRP inquiries. For example, you if you are placing demand for Order # 100, Line #2, you could populate this field with a value of 2. The DEMAND_SOURCE_LINE could still be a unique internal identifier, but the displayed value would be more meaningful to the user.

USER_DELIVERY

You can populate this column much the same way as you use USER_LINE_NUM. It is not, however, displayed on any standard inquiries or reports.

EXTERNAL_SOURCE_CODE

You can use this column as an external system reference. The values entered are transferred directly to the transaction history table.

EXTRENAL_SOURCE_LINE_ID

You can use this column as an external system reference. The values entered are transferred directly to the transaction history table.

ACTION_CODE

Enter the code for the action to be performed on each row. The valid codes are:

Code	Action
100	ATP Check
110	Demand Add
130	Reservation Add
140	Reservation Modify
150	Demand and Reservation Add
410	Update Forecast Attributes
420	Pick Release
610	ATS Query
620	ATS Quick Pick

Table 5 – 4 Action Codes (Page 1 of 1)

Validation

Oracle Inventory lets you choose the level of validation you want performed against interfaced demand rows. Using the `VALIDATION_REQUIRED` flag, you can specify whether you want full validation or only partial validation of columns required for derivation of other required columns. For example, `ORGANIZATION_ID` is always validated because there are dependent attributes such as `LOCATOR_ID` that require a valid organization for derivation. `REVISION`, on the other hand, has no dependencies, and therefore is not validated if the `VALIDATION_REQUIRED` flag is not set.

For more information on this subject, consult the details for the column `ERROR_CODE` in the `MTL_DEMAND_INTERFACE` interface table. You can find this information in the *Oracle Inventory Technical Reference Manual*.



Table and View Definitions, Detailed Design
(*Oracle Inventory Technical Reference Manual*)

Resolving Failed Demand Interface Rows

Viewing Failed Transactions

You can view both pending and failed demand rows in the MTL_DEMAND_INTERFACE table using the Oracle Inventory View Pending Interface Activity form. If your requests errored out and you would like to resubmit them, you can do so using this form. If you set 'Resubmit=Yes', the interface processing flags will automatically be reset so the Demand Manager will pick them up.

Fixing Failed Transactions Options

Errors in the interface may be caused by problems unrelated to your request. For example, there may be validation that failed because an entity that was being checked had the wrong status (for example, disabled), or the failure could even be the result of a system error, such as running out of space. In these cases, it may be acceptable to simply resolve the conflict and resubmit the same interfaced rows by either using the View Pending Interface Activity form to resubmit your requests, or by directly updating the PROCESS_FLAG and LOCK_FLAG values via SQL*PLUS.

If, however, you need to make changes to the source information because of invalid data, you need to either delete the failed transactions and resubmit them from the feeder system, or update the transaction in the interface table using SQL*PLUS. When you resubmit updated transactions for processing, all validation is performed again.

Open Transaction Interface

Oracle Inventory provides an open interface for you to easily load transactions from external applications and feeder systems. These transactions could include sales order shipment transactions from an order entry system other than Oracle Order Entry, or they could be simple material issues, receipts, or transfers loaded from data collection devices. The following transaction types are supported by this interface:

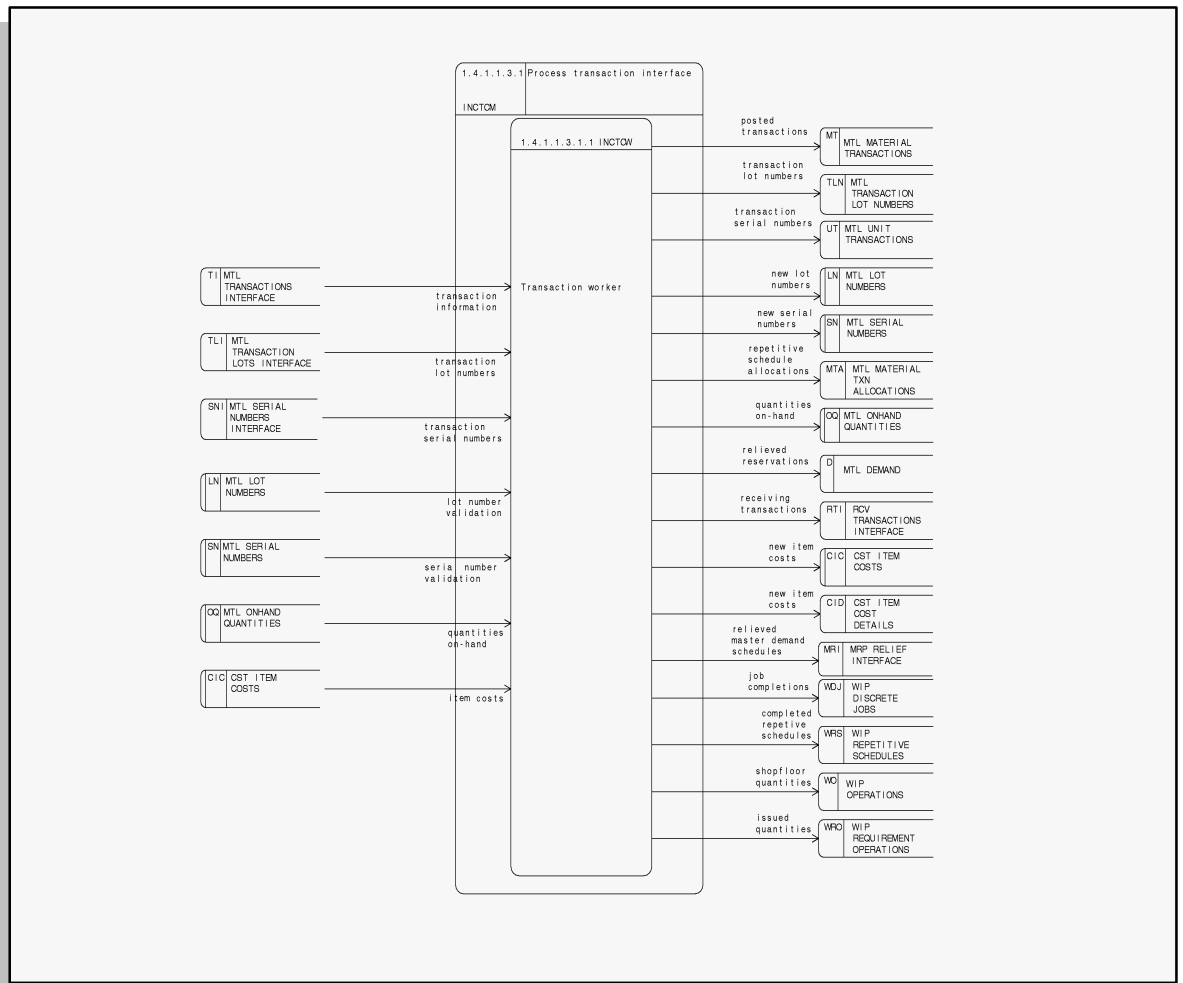
- Inventory issues and receipts (including user-defined transaction types)
- Subinventory transfers
- Direct inter-organization transfers
- Intransit shipments
- WIP component issues and returns
- WIP assembly completions and returns
- Sales order shipments
- Inventory average cost updates

This interface is also used as an integration point with Oracle Order Entry for shipment transactions. Oracle Order Entry's Inventory Interface program populates the interface tables with transactions submitted through the Confirm Shipments form.

Functional Overview

The following data flow diagram shows the key tables and programs that comprise the Open Inventory Transaction Interface for Inventory Movement Transactions, WIP Issue and Completion Transactions, Sales Order Shipments, and Inventory Average Cost Update Transactions.

Figure 5 – 2



You must write the load program that inserts a single row for each transaction into the MTL_TRANSACTIONS_INTERFACE table. For material movement of items that are under lot or serial control, you must also insert rows into MTL_TRANSACTION_LOTS_INTERFACE and MTL_SERIAL_NUMBERS_INTERFACE respectively.

There are two modes you can use to process your transactions through the interface. In the first processing mode, you populate the interface table only. Then the Transaction Manager polls the interface table

asynchronously looking for transactions to process, groups the transaction rows, and launches a Transaction Worker to process each group. In the second processing mode, you insert the rows in the interface table and call a Transaction Worker directly, passing the group identifier of the interfaced transactions as a parameter so that the worker can recognize which subset of transactions to process.

The Transaction Worker calls the Transaction Validator which validates the row, updates the error code and explanation if a validation or processing error occurs, and derives or defaults any additional columns.

Next, the Transaction Processor records the transaction details in the transaction history table along with relevant current cost information. All material movement transactions update inventory perpetual balances for the issue, receipt, or transfer locations.

Once the transaction has been successfully processed, the corresponding row is deleted from the interface table. Finally, the transaction is costed by the transaction cost processor which runs periodically, picking up all transactions from the history table that have not yet been marked as 'costed'.

Additional Transaction Processing Flow Steps

The following transactions require additional processing by the transaction processor or other modules.

Inventory Issue Transactions

Inventory Issue transactions consume any existing reservations where the Transaction Source Type and Source match. For example, if you reserved 10 boxes of paper for the Finance department, and then you issue 4 boxes to that department, the reservation will automatically be partially consumed, with a remaining balance of 6 reserved boxes.

Average Cost Transactions

In average cost organizations, receipts and average cost update transactions modify the item's average cost using the current average cost, on-hand quantity, and the transaction value and quantity (if appropriate) to calculate the new average.

WIP Issue Transactions

WIP issue transactions also update quantity issued for all material requirements on the job or repetitive schedule and charge the costs of issued components to the job/schedule.

WIP Completion Transactions

WIP completion transactions update the job or repetitive schedule completed quantities, launch appropriate backflush transactions, and relieve costs of completed assembly from the job/schedule. If you are completing an ATO assembly, you must specify the sales order demand details so that Oracle Inventory can reserve the completed units to the appropriate sales order line/shipment.

Sales Order Shipment Transactions

For sales order shipment transactions, the Transaction Processor attempts to consume any reservations that may have been created for an order by matching the Order, Line, Delivery, and Picking Line identifiers. If MRP is installed, the processor also creates an interface row in MRP_RELIEF_INTERFACE that the MRP Planning Manager uses to relieve the Master Demand Schedule.

Lot and Serial Transaction Detail Relationships

If you are transacting items under lot and/or serial control, you need to link the lot/serial transaction detail rows to their parent row. You accomplish this by populating MTL_TRANSACTIONS_INTERFACE.TRANSACTION_INTERFACE_ID with a unique value to be used as the primary key to link the child lot/serial rows. If the item is under lot control, you populate the foreign key MTL_TRANSACTION_LOTS_INTERFACE.TRANSACTION_INTERFACE_ID with the same value for all child lot rows of the transaction and ensure that the total of all the lot quantities adds up to the transaction quantity on the parent row. Similarly, if the item is under serial control, you populate the foreign key MTL_SERIAL_NUMBERS_INTERFACE.TRANSACTION_INTERFACE_ID with the value in the parent row and ensure that the total number of serial numbers adds up to the transaction quantity of the parent row.

If the item is under both lot and serial control, the serial interface rows must belong to lot parent rows. This means that the relationship between MTL_TRANSACTIONS_INTERFACE and MTL_TRANSACTION_LOT_NUMBERS remains the same as in the case where the item is only under lot control, but you also need to populate each lot row with a unique value in MTL_TRANSACTION_LOT_NUMBERS.SERIAL_TRANSACTION_TEMP_ID. You then need to populate the foreign key MTL_SERIAL_NUMBERS_INTERFACE.TRANSACTION_INTERFACE_ID with the value in the parent lot row and ensure that the total number of serial numbers adds up to the lot quantity in the parent row.

Setting Up the Transaction Interface

Setting Up the Inventory Concurrent Manager

For optimal processing in the Inventory Transaction Interface, you need to set up your concurrent manager to best handle your transaction volumes while balancing your performance requirements and your system load restrictions. Oracle Inventory ships the Transaction Manager to be run in Inventory's own concurrent manager named Inventory Manager. It is defaulted to run in the Standard workshift with Target Processes = 1 and Sleep Time of 60 seconds.

With this configuration, the Material Transaction Manager and all Transaction Workers that are spawned must share the same processing queue. If you have the available resources, you can substantially reduce the time to process your interfaced transactions by increasing the target processes and reducing the concurrent manager sleep time using the Define Concurrent Managers form. This will allow Transaction Workers to run in parallel with the Transaction Manager and with each other.

Starting the Inventory Transaction Manager

Once you have set up the Inventory concurrent manager, you can launch the Inventory Transaction Manager in the Oracle Inventory's Request Interface Managers form. This launches the Material Transaction manager and lets you specify the polling interval and the number of transactions to be processed by each worker. After polling the MTL_TRANSACTIONS_INTERFACE table for eligible rows, the Transaction Manager creates the necessary number of Transaction Workers to process the load.

Submitting a Transaction Worker Directly as a Concurrent Process

The transaction worker can be directly called either from an Oracle Form or a c program. You can also launch a worker from the operating system using the Application Object library CONCSUB utility. You need to specify the following parameters in the given order.

HEADER_ID	This is the transaction_header_id that you want the worker to process. If no header id is passed the worker will assign itself.
TABLE	Pass '1' for the Interface table and '2' for the temp table.

SOURCE _HEADER_ID	This column will be used to select rows to process if HEADER_ID is not specified.
SOURCE_CODE	This column is used to select rows to process if header id is not specified.

Setting Up Your Sales Order Flexfield

Oracle Inventory uses a flexfield to hold the unique Sales Order name so that it does not need to join back to the feeder Order Entry system. This means that you must set up Inventory's Sales Order flexfield (MKTS) using the Define Key Flexfield Segments form with enough segments so that the combination is unique across all orders.

For example, Oracle Order Entry guarantees uniqueness within an installation, order type, and order number. Consequently, standard installation steps require that you set up a three segment. If you can guarantee that one segment is sufficient (for example, Order Number), then that is all you need to enable in your flexfield definition.

When you enter shipment transactions into the interface, you should use the Sales Order segment values to identify the order. The Material Transaction Manager will validate against MTL_SALES_ORDERS, and if the code combination does not already exist will create a new one. All references to the order number internal to Inventory in reports and inquiries will be based on this relationship.

Inserting into the Transaction Interface Tables

This section provides a chart for each interface table that lists all columns, followed by a section giving a brief description of a subset of columns requiring further explanation. The chart identifies each column's datatype and whether it is Required, Derived, or Optional. Many of the columns are conditionally required. Reference numbers corresponding to notes immediately following the table help identify the mandatory conditions.

Several of the attributes in the interface tables can be populated using either the user-friendly values or the internal identifiers. This is particularly true of flexfields, such as Item, Locator, and Distribution Account. In these cases, you have the option to specify either the flexfield segment representation or the internal identifier (for example, INVENTORY_ITEM_ID) for the required value.

If you populate the user-friendly values, the Transaction Validator will automatically validate them and derive the internal identifiers. If the

translation is already available to the external system, it may be advantageous to use the internal identifiers to improve performance (see discussion below on validation).

MTL_TRANSACTIONS_INTERFACE

The following graphic describes the
MTL_TRANSACTIONS_INTERFACE Interface table:

MTL_TRANSACTIONS_INTERFACE Column Name	Type	Required	Derived	Optional
SOURCE_CODE	Varchar2(30)	✓		
SOURCE_LINE_ID	Number	✓		
SOURCE_HEADER_ID	Number	✓		
PROCESS_FLAG	Number(1)	✓		
TRANSACTION_MODE	Number	✓		
LOCK_FLAG	Number(1)			✓
TRANSACTION_HEADER_ID	Number		✓	
ERROR_CODE	Varchar2(240)		✓	
ERROR_EXPLANATION	Varchar2(240)		✓	
VALIDATION_REQUIRED	Number			✓
TRANSACTION_INTERFACE_ID	Number	✓		
INVENTORY_ITEM_ID	Number	✓		
ITEM_SEGMENT1 to ITEM_SEGMENT20	Varchar2(40)	✓		
REVISION	Varchar2(3)	1		
ORGANIZATION_ID	Number	✓		
SUBINVENTORY_CODE	Varchar2(10)	2		
LOCATOR_ID	Number	3		
LOC_SEGMENT1 to LOC_SEGMENT20	Varchar2(40)	3		
TRANSACTION_QUANTITY	Number	✓		

Table 5 – 5 Transaction Interface Table (Page 1 of 4)

MTL_TRANSACTIONS_ INTERFACE				
Column Name	Type	Required	Derived	Optional
TRANSACTION_UOM	Varchar2(3)	✓		
PRIMARY_QUANTITY	Number		✓	
TRANSACTION_DATE	Date	✓		
ACCT_PERIOD_ID	Number		✓	
TRANSACTION_SOURCE_ID	Number	✓		
DSP_SEGMENT1toDSP_SEGMENT30	Varchar2(40)	✓		
TRANSACTION_SOURCE_NAME	Varchar2(30)	✓		
TRANSACTION_SOURCE_TYPE_ID	Number		✓	
TRANSACTION_ACTION_ID	Number		✓	
TRANSACTION_TYPE_ID	Number	✓		
REASON_ID	Number			✓
TRANSACTION_REFERENCE	Varchar2(240)			✓
TRANSACTION_COST	Number	4		
DISTRIBUTION_ACCOUNT_ID	Number	5		
DST_SEGMENT1toDST_SEGMENT30	Varchar2(25)	5		
CURRENCY_CODE	Varchar(30)			✓
CURRENCY_CONVERSION_TYPE	Varchar(30)			✓
CURRENCY_CONVERSION_RATE	Number			✓
CURRENCY_CONVERSION_DATE	Date			✓
USSGL_TRANSACTION_CODE	Varchar(30)			✓
ENCUMBRANCE_ACCOUNT	Number			✓
ENCUMBRANCE_AMOUNT	Number			✓
VENDOR_LOT_NUMBER	Varchar2(30)			✓
TRANSFER_SUBINVENTORY	Varchar2(10)	6		

Table 5 – 5 Transaction Interface Table (Page 2 of 4)

MTL_TRANSACTIONS_ INTERFACE Column Name	Type	Required	Derived	Optional
TRANSFER_ORGANIZATION	Number	6		
TRANSFER_LOCATOR	Number	3,6		
XFER_LOC_SEGMENT1toX- FER_LOC_SE	Varchar2(40)	3,6		
SHIPMENT_NUMBER	Varchar2(30)	7		
TRANSPORTATION_COST	Number			✓
TRANSPORTATION_ACCOUNT	Number			✓
TRANSFER_COST	Number			✓
FREIGHT_CODE	Varchar2(25)			✓
CONTAINERS	Number			✓
WAYBILL_AIRBILL	Varchar2(20)			✓
EXPECTED_ARRIVAL_DATE	Date			✓
NEW_AVERAGE_COST	Number	8		
VALUE_CHANGE	Number	8		
PERCENTAGE_CHANGE	Number	8		
DEMAND_ID	Number			9
PICKING_LINE_ID	Number			
DEMAND_SOURCE_HEADER_ID	Number			10
DEMAND_SOURCE_LINE	Varchar2(30)			10
DEMAND_SOURCE_DELIVERY	Varchar(30)			10
WIP_ENTITY_TYPE	Number	11,12		
SCHEDULE_ID	Number		11,12	
OPERATION_SEQ_NUM	Number	11	12	
REPETITIVE_LINE_ID	Number	13		
NEGATIVE_REQ_FLAG	Number			✓
TRX_SOURCE_LINE_ID	Number			9

Table 5 – 5 Transaction Interface Table (Page 3 of 4)

MTL_TRANSACTIONS_ INTERFACE Column Name	Type	Required	Derived	Optional
TRX_SOURCE_DELIVERY_ID	Number			9
CUSTOMER_SHIP_ID	Number			✓
SHIPPABLE_FLAG	Varchar2(1)		✓	
LAST_UPDATE_DATE	Date	✓		
LAST_UPDATED_BY	Number	✓		
CREATION_DATE	Date	✓		
CREATED_BY	Number	✓		
LAST_UPDATE_LOGIN	Number			✓
REQUEST_ID	Number			✓
PROGRAM_APPLICATION_ID	Number			✓
PROGRAM_ID	Number			✓
PROGRAM_UPDATE_DATE	Date			✓
ATTRIBUTE_CATEGORY	Varchar2(30)			✓
ATTRIBUTE1 to ATTRIBUTE15	Varchar2(150)			✓

Table 5 – 5 Transaction Interface Table (Page 4 of 4)

1. If under revision control
2. All transaction types except average cost update
3. If under locator control
4. Inventory Issues and Receipts in an average cost organization
5. Inventory Issues/Receipts of an asset item to/from an asset subinventory and sales order shipment transactions
6. Inventory direct transfers (inter- or intra-organization)
7. Intransit shipments
8. Average cost update transactions only
9. Sales order shipment transactions
10. To reserve/unreserve ATO items to a sales order upon completion/return from a WIP job

11. WIP component issues/returns
12. WIP assembly completions/returns
13. Repetitive schedules

SOURCE_CODE**REQUIRED**

This column is required for Sales Order transactions to identify the source Order Entry system. For other transaction types, you can enter any useful value for tracking purposes. The values entered are transferred directly to the transaction history table.

SOURCE_HEADER_ID**REQUIRED**

You can use this column as an external system reference. The values entered are transferred directly to the transaction history table.

SOURCE_LINE_ID**REQUIRED**

You can use this column as an external system reference. The values entered are transferred directly to the transaction history table.

PROCESS_FLAG**REQUIRED**

This column controls whether rows in the interface table are processed or not. You should insert a row that you intend to be processed with a value of 1 (Yes). The valid values are:

- 1 – Yes
- 2 – No
- 3 – Error

TRANSACTION_MODE**REQUIRED**

This column determines how the interfaced transactions will be processed. The valid options are:

- 2 – Concurrent
- 3 – Background

Interface transactions marked for 'Background' processing will be picked up by the transaction manager polling process and assigned to a transaction worker. These transactions will not be processed unless the transaction manager is running.

You use 'Concurrent' transaction mode if you want to launch a dedicated transaction worker to explicitly process a set of transactions. The Transaction Manager will not process transactions marked for concurrent processing.

<u>LOCK_FLAG</u>	<u>DERIVED</u>
-------------------------	-----------------------

The Transaction Manager uses this column to manage the worker assignment process. You should only need to update this column if a transaction has failed due to an exceptional failure such as the system going down in the middle of transaction worker processing. In this case, you will need to reset the LOCK_FLAG to 2 so your failed transactions can be reprocessed.

<u>TRANSACTION_HEADER_ID</u>	<u>DERIVED, REQUIRED</u>
-------------------------------------	---------------------------------

This column groups transactions for assignment to specific transaction workers. Depending on the value of TRANSACTION_MODE, this column is either required (concurrent mode) or derived by the transaction manager (background mode). This column maps to MTL_MATERIAL_TRANSACTIONS.TRANSACTION_SET_ID in the transaction history tables.

<u>ERROR_CODE</u>	<u>DERIVED</u>
--------------------------	-----------------------

If a transaction error occurs, the Transaction Validator populates this column with short descriptive text indicating the type of error that has occurred.

<u>ERROR_EXPLANATION</u>	<u>DERIVED</u>
---------------------------------	-----------------------

If a transaction error occurs, the Transaction Validator populates this column with an explanation of the error. If an explanation is not provided, check the log file for details using the View Requests form.

<u>VALIDATION_REQUIRED</u>	<u>OPTIONAL</u>
-----------------------------------	------------------------

You can use this flag to control whether the Transaction Validator skips certain validation steps for certain transaction types. The options are:

- 1 – Full validation
 - 2 – Validate only columns required for derivation
- If you leave this field null, Full validation is used.



Validation (See page 5 – 44)

TRANSACTION_INTERFACE_ID	CONDITIONAL
---------------------------------	--------------------

This column is required for transactions of items under lot or serial control. The value in the column in this table is used to identify the child rows in the lot or serial interface tables

MTL_TRANSACTION_LOTS_INTERFACE and
MTL_SERIAL_NUMBERS_INTERFACE.

If the transacted item is under lot control, this column maps to
MTL_TRANSACTION_LOTS_INTERFACE.
TRANSACTION_INTERFACE_ID. If the transacted item is under
serial control and not lot control, this column maps to
MTL_SERIAL_NUMBERS_INTERFACE.
TRANSACTION_INTERFACE_ID.

TRANSACTION_QUANTITY	REQUIRED
-----------------------------	-----------------

Enter the transaction quantity in the transaction unit of measure. The quantity should be positive for receipts into inventory, and negative for both issues out of inventory and transfers. Enter a quantity of 0 for Average Cost Update transactions.

TRANSACTION_UOM	REQUIRED
------------------------	-----------------

You can enter the TRANSACTION_QUANTITY in any unit of measure that has conversion rates defined to the item's primary unit of measure. Use this column to specify the transacted unit of measure even if it is the same as the primary unit of measure.

PRIMARY_QUANTITY	DERIVED
-------------------------	----------------

This column is the transaction quantity in the item's primary unit of measure calculated using TRANSACTION_QUANTITY and
TRANSACTION_UOM.

ACCT_PERIOD_ID	DERIVED
-----------------------	----------------

This column is derived using the entered TRANSACTION_DATE to determine within which period the transaction occurred. The transaction date must be on or before the system date at time of transaction processing, and the transaction date must lie within the boundaries of an open period (in ORG_ACCT_PERIODS).

TRANSACTION_TYPE_ID**REQUIRED**

Enter the type of transaction you are executing. The transaction types and internal IDs supported by the Open Interface are:

Transaction Type	Internal ID
Account Issue	01
Account Alias Issue	31
Miscellaneous Issue	32
Issue Components to WIP	35
Return Assemblies to WIP	17
Account Receipt	40
Account Alias Receipt	41
Miscellaneous Receipt	42
Return Components from WIP	43
WIP Assembly Completion	44
Subinventory Transfer	02
Direct Inter-Organization Transfer	03
Intransit Shipment	21
Average Cost Update	80
Sales Order Shipment	33

Table 5 – 6 Transaction Types and Internal IDs (Page 1 of 1)

You can identify the TRANSACTION_TYPE_ID for user-defined transactions by selecting from MTL_TRANSACTION_TYPES where TRANSACTION_TYPE_NAME is the transaction type you wish to use.

TRANSACTION_SOURCE_TYPE_ID**DERIVED**

This column is derived from MTL_TRANSACTION_TYPES using the value you enter in TRANSACTION_TYPE_ID.

TRANSACTION_SOURCE_NAME **CONDITIONAL**

This column is required for user-defined transaction source types. Enter the value of the source name, such as an order number, to be displayed on all transaction reports and inquiries.

TRANSACTION_SOURCE_ID **CONDITIONAL**

TRANSACTION_SOURCE_ID or the corresponding flexfield segment columns (DSP_SEGMENT1 to DSP_SEGMENT30) are required for all transaction source types other than those that are user-defined. You should enter the foreign key ID that points to the context table identified by the transaction source type.

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

Table 5 – 7 TRANSACTION_SOURCE_ID, Foreign Key References (Page 1 of 1)

DSP_SEGMENT1 TO DSP_SEGMENT30 **CONDITIONAL**

You can use these flexfield segment columns instead of TRANSACTION_SOURCE_ID to enter the more user-friendly information. For example, if the interfaced transaction is for an Issue to Account transaction type, you would enter the GL Code Combination segment values in these columns instead of putting the Code GL Code Combination ID in TRANSACTION_SOURCE_ID.

TRANSACTION_ACTION_ID **DERIVED**

This column is derived from MTL_TRANSACTION_TYPES using the value you enter in TRANSACTION_TYPE_ID.

OPERATION_SEQ_NUM **CONDITIONAL**

For assembly completions and returns, this value is derived. For WIP component issues and returns with routings, this value is required. For WIP routings, enter 1.

<u>WIP_ENTITY_TYPE</u>	<u>REQUIRED</u>
-------------------------------	------------------------

For WIP component issues and returns, and WIP assembly completions and returns, enter one of the following values:

- 1 – Standard discrete jobs
- 2 – Repetitive schedules
- 3 – Non-standard discrete jobs

<u>REASON_ID</u>	<u>OPTIONAL</u>
-------------------------	------------------------

Use this column to specify a transaction reason from the predefined list of reasons in MTL_TRANSACTION_REASONS.

<u>TRANSACTION_REFERENCE</u>	<u>OPTIONAL</u>
-------------------------------------	------------------------

Use this column to enter any transaction reference information. Oracle Inventory displays this column on transaction inquiries and reports.

<u>TRANSACTION_COST</u>	<u>OPTIONAL</u>
--------------------------------	------------------------

You can use this column to specify a transaction unit cost for average cost Inventory issues and receipts. If you leave it blank, the current system unit cost will be used.

<u>DISTRIBUTION_ACCOUNT_ID</u>	<u>CONDITIONAL, DERIVED</u>
---------------------------------------	------------------------------------

Use this column (or the flexfield segment columns) to specify the account to charge for the cost of the Inventory transaction. It is required for user-defined transactions, and derived by the Transaction Worker based on the transaction source type and source for Account Issue/Receipt and Account Alias Issue/Receipt transactions.

<u>DST_SEGMENT1 TO DST_SEGMENT30</u>	<u>CONDITIONAL</u>
---	---------------------------

You can use these flexfield segment columns instead of DISTRIBUTION_ACCOUNT_ID to enter the more user-friendly information. For example, if the interfaced transaction is for an Issue to Account transaction type, you would enter the GL Code Combination segment values in these columns instead of putting the Code GL Code Combination ID in DISTRIBUTION_ACCOUNT_ID.

CURRENCY_CODE	OPTIONAL
----------------------	-----------------

If your transaction cost is in a different currency than the functional currency of your set of books, enter the currency code.

CURRENCY_CONVERSION_TYPE	OPTIONAL
---------------------------------	-----------------

If you enter a currency code other than the functional currency for your set of books, enter the conversion type.

CURRENCY_CONVERSION_RATE	OPTIONAL
---------------------------------	-----------------

If you enter a currency code other than the functional currency for your set of books, enter the conversion rate

CURRENCY_CONVERSION_DATE	OPTIONAL
---------------------------------	-----------------

Enter the currency conversion date for which the conversion rate is valid for the transaction.

VENDOR_LOT_NUMBER	OPTIONAL
--------------------------	-----------------

Use this column as transaction reference information and/or to cross-reference against internal lot numbers.

TRANSFER_ORGANIZATION	CONDITIONAL
------------------------------	--------------------

This column is required for all inter-organization transfers. Enter the destination organization's internal ID.

TRANSFER_SUBINVENTORY	CONDITIONAL
------------------------------	--------------------

This column is required for subinventory transfers within the same organization and direct transfers from one organization to another. For these scenarios, enter the destination subinventory.

TRANSFER_LOCATOR	CONDITIONAL
-------------------------	--------------------

This column is required for subinventory transfers within the same organization and direct transfers from one organization to another when the item being transferred is under locator control in the destination subinventory. For these scenarios, enter the destination locator internal ID.

XFER_LOC_SEGMENT1-XFER_LOC_SEGMENT20 CONDITIONAL

When a transfer locator is required, you can optionally use these columns instead of TRANSFER_LOCATOR when you want to use the user-friendly flexfield representation of the transfer locator instead of the internal ID.

SHIPMENT_NUMBER **CONDITIONAL**

This column is required for intransit shipments. It groups shipment lines in RCV_SHIPMENT_LINES under a parent shipment number in RCV_SHIPMENT_HEADERS.

The Transaction Worker will not process intransit transactions if a shipment header already exists in RCV_SHIPMENT_HEADERS that matches SHIPMENT_NUMBER. If you want to group shipment lines under the same header, you must ensure they are processed by the same worker. You can accomplish this using the concurrent processing mode, using the TRANSACTION_HEADER_ID to group your interface transactions, and directly calling a Transaction Worker to process that group.

NEW_AVERAGE_COST **CONDITIONAL**

Average cost update transactions require that either NEW_AVERAGE_COST, VALUE_CHANGE, or PERCENTAGE_CHANGE be populated, depending on the type of cost update being performed.

VALUE_CHANGE **CONDITIONAL**

See NEW_AVERAGE_COST.

PERCENTAGE_CHANGE **CONDITIONAL**

See NEW_AVERAGE_COST.

DEMAND_ID **OPTIONAL**

Use this column for sales order shipment transactions to identify the exact reservation row to be relieved in MTL_DEMAND. If you do not have the DEMAND_ID information, leave this column blank, and the Transaction Processor will try to match reservations to relieve by checking MTL_DEMAND to see if there are any reservations where there is a match on:

MTL_TRANSACTIONS_INTERFACE MTL_DEMAND	
ORGANIZATION_ID	ORGANIZATION_ID
INVENTORY_ITEM_ID	INVENTORY_ITEM_ID
TRANSACTION_SOURCE_TYPE_ID	DEMAND_SOURCE_TYPE_ID
TRANSACTION_SOURCE_ID	DEMAND_SOURCE_HEADER_ID
TRANSACTION_SOURCE_LINE_ID	DEMAND_SOURCE_LINE_ID
TRANSACTION_SOURCE	DEMAND_SOURCE
DELIVERY_ID	DELIVERY_ID

Table 5 – 8 Table Mapping: MTL_TRANSACTIONS_INTERFACE to MTL_DEMAND (Page 1 of 1)

TRX_SOURCE_LINE_ID **OPTIONAL**

Use this column to specify details of reservations to be relieved with an issue transaction. See DEMAND_ID.

TRX_SOURCE_DELIVERY_ID **OPTIONAL**

Use this column to specify details of reservations to be relieved with an issue transaction. See DEMAND_ID.

DEMAND_SOURCE_HEADER_ID

Use this column for completion (and returns) of ATO items from a Final Assembly Order if the quantity you are completing is to be reserved to an existing sales order. Enter values in DEMAND_SOURCE_HEADER_ID, DEMAND_SOURCE_LINE_ID, and DEMAND_SOURCE_DELIVERY_ID that match the appropriate demand rows in MTL_DEMAND. The transaction processor will automatically create a reservation for the completed quantity to that sales order.

DEMAND_SOURCE_LINE_ID

See DEMAND_SOURCE_HEADER_ID.

DEMAND_SOURCE_DELIVERY_ID

See DEMAND_SOURCE_HEADER_ID.

MTL_TRANSACTION_LOTS_INTERFACE

The following graphic describes the
MTL_TRANSACTION_LOTS_INTERFACE Interface table:

MTL_TRANSACTION_LOTS_INTERFACE Column Name	Type	Required	Derived	Optional
TRANSACTION_INTERFACE_ID	Number	✓		
SOURCE_CODE	Varchar2(30)			✓
SOURCE_LINE_ID	Number			✓
LOT_NUMBER	Varchar2(30)	✓		
LOT_EXPIRATION_DATE	Date	1		
TRANSACTION_QUANTITY	Number	✓		
PRIMARY_QUANTITY	Number		✓	
SERIAL_TRANSACTION_TEMP_ID	Number	2		
ERROR_CODE	Varchar2(30)		✓	
LAST_UPDATE_DATE	Date	✓		
LAST_UPDATED_BY	Number	✓		
CREATION_DATE	Date	✓		
CREATED_BY	Number	✓		
LAST_UPDATE_LOGIN	Number			✓
REQUEST_ID	Number			✓
PROGRAM_APPLICATION_ID	Number			✓
PROGRAM_ID	Number			✓
PROGRAM_UPDATE_DATE	Date			✓

Table 5 – 9 Transaction Lot Numbers Interface Table (Page 1 of 1)

1. If item is under lot expiration control
2. If item is under both lot and serial control

LOT_NUMBER**REQUIRED**

Enter the lot number that is being transacted.

TRANSACTION_INTERFACE_ID**REQUIRED**

Use this column to associate lot transaction detail rows with the parent transaction row in MTL_TRANSACTIONS_INTERFACE.

SERIAL_TRANSACTION_TEMP_ID**CONDITIONAL**

This column is required only for items under both lot and serial control. It is used to identify the child rows in MTL_SERIAL_NUMBERS_INTERFACE.

MTL_SERIAL_NUMBERS_INTERFACE

The following graphic describes the MTL_SERIAL_NUMBERS_INTERFACE Interface table:

MTL_SERIAL_NUMBERS_INTERFACE Column Name	Type	Required	Derived	Optional
TRANSACTION_INTERFACE_ID	Number	✓		
SOURCE_CODE	Varchar2(30)			✓
FM_SERIAL_NUMBER	Varchar2(30)	✓		
TO_SERIAL_NUMBER	Varchar2(30)			✓
SOURCE_LINE_ID	Number			✓
VENDOR_SERIAL_NUMBER	Varchar2(30)			✓
ERROR_CODE	Varchar2(30)		✓	
LAST_UPDATE_DATE	Date	✓		
LAST_UPDATED_BY	Number	✓		
CREATION_DATE	Date	✓		
CREATED_BY	Number	✓		
LAST_UPDATE_LOGIN	Number			✓

Table 5 – 10 Transaction Serial Numbers Interface Table (Page 1 of 2)

MTL_SERIAL_NUMBERS_ INTERFACE Column Name	Type	Required	Derived	Optional
REQUEST_ID	Number			✓
PROGRAM_APPLICATION_ID	Number			✓
PROGRAM_ID	Number			✓
PROGRAM_UPDATE_DATE	Date			✓

Table 5 – 10 Transaction Serial Numbers Interface Table (Page 2 of 2)

FM_SERIAL_NUMBER **REQUIRED**

Enter the starting serial number in the range. If you only enter the 'from' serial number, the Transaction Processor assumes that only 1 serial number is being transacted.

TO_SERIAL_NUMBER **OPTIONAL**

You can enter a 'to' serial number to specify a range. The transaction processor will attempt to transact all serial numbers within the range of the rightmost numeric digits.

TRANSACTION_INTERFACE_ID **REQUIRED**

Use this column to associate serial number transaction detail rows with their parent rows. If the item is under both lot and serial control, this should point to MTL_TRANSACTION_LOTS_INTERFACE SERIAL_TRANSACTION_TEMP_ID. Otherwise, it should point to MTL_TRANSACTIONS_INTERFACE. TRANSACTION_INTERFACE_ID

VENDOR_SERIAL_NUMBER **OPTIONAL**

You can use this column to enter vendor cross-reference information. The vendor serial number is stored in the serial number table MTL_SERIAL_NUMBERS.

Validation

Oracle Inventory lets you choose the level of validation you want performed against interfaced transaction rows. Using the `VALIDATION_REQUIRED` flag, you can specify whether you want full validation or only partial validation of columns required for derivation of other required columns. For example, `ORGANIZATION_ID` is always validated because there are dependent attributes such as `LOCATOR_ID` that require a valid organization for derivation. `REVISION`, on the other hand, has no dependencies, and therefore is not validated if the `VALIDATION_REQUIRED` flag is not set.

The validation and derivation processes will provide an error code and description for all transaction rows that fail explicit validation checks. If an error occurs during reservation relief for a specific transaction, all rows in the transaction processing group will be errored out with a common error message. This should only happen, however, if there is an Oracle error or table deadlock during processing.

If an error occurs in the transaction processor, the entire transaction processing group is marked with the error code, while the transaction row(s) that actually failed will have an error explanation.

Resolving Failed Transaction Interface Rows

Viewing Failed Transactions

You can view both pending and failed Inventory transactions in the `MTL_TRANSACTIONS_INTERFACE` table using the Oracle Inventory View Pending Interface Activity form. If your transactions errored out and you would like to resubmit them, you can do so using this form. If you set 'Resubmit=Yes', the interface processing flags will automatically be reset so the Transaction Manager will pick them up.

Fixing Failed Transactions Options

Errors in the interface may be caused by problems unrelated to your interfaced transactions. For example, there may be validation that failed because an entity that was being checked had the wrong status (for example, disabled), or the failure could even be the result of a system error, such as running out of space. In these cases, it may be acceptable to simply resolve the conflict and resubmit the same interfaced rows by either using the View Pending Interface Activity

form to resubmit your transactions, or by directly updating the PROCESS_FLAG and LOCK_FLAG values via SQL*PLUS.

If, however, you need to make changes to the transaction data itself, you need to either delete the failed transactions and resubmit them from the feeder system, or update the transaction in the interface table using SQL*PLUS. When you resubmit updated transactions for processing, all validation is performed again.

Open Replenishment Interface

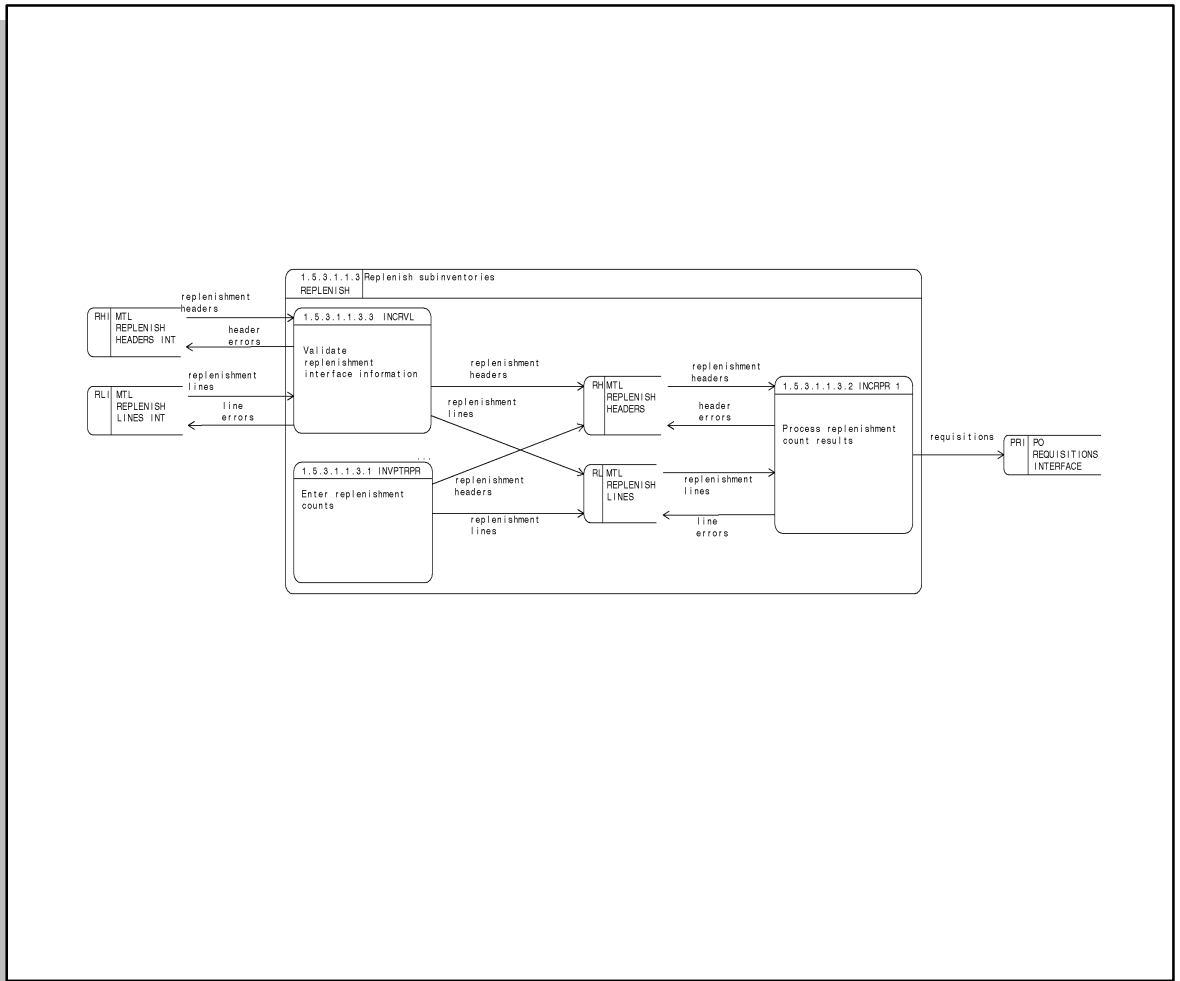
Oracle Inventory provides an open interface for you to easily load replenishment requests from external systems such as a bar-code application. Such requests may be in the form of stock-take counts or requisition requests for subinventories in which you do not track quantities.

You may also use the Open Replenishment Interface to process requisition requests generated by external applications for tracked subinventories.

Functional Overview

The following data flow diagram shows the key tables and programs that comprise the Open Replenishment Interface:

Figure 5 – 3



You must write the load program that inserts a single row for each replenishment count/request into the MTL_REPLENISH_HEADERS_INT table. A record for each item included in the count header must be inserted into the MTL_REPLENISH_LINES_INT table.

There are two modes you can use to process your replenishment counts through the interface. These are 'Concurrent' and 'Background' modes.

Under Concurrent mode processing, you populate the interface tables for a specific replenishment count and then call the replenishment validator from the Oracle Inventory menu. The validator processes the

replenishment count specified as a parameter at process submission, validating rows in both the MTL_REPLENISH_HEADER_INT and MTL_REPLENISH_LINES_INT tables. The validator derives any additional columns and updates the error flag if an error is detected.

For Background mode processing, you populate the interface tables and then let the Replenishment Validator asynchronously poll the tables for replenishment counts to process.

If the replenishment count, both header and lines, passes all required validation, the records are inserted into the MTL_REPLENISH_HEADERS and MTL_REPLENISH_LINES tables and are deleted from the interface tables. If an error is detected during the validation process, the header and corresponding replenishment lines will be left in the interface table.

Once the lines are in the internal replenishment tables, you use the Replenishment Processor as described in the Oracle Inventory Reference Manual to process the counts and create requisitions.

Setting Up the Replenishment Interface

If you want to process your replenishment counts in Background mode, you must submit the Replenishment Validator with 'Background' specified as the Processing Mode parameter. This may be done from the Standard Report Submission form on the Oracle Inventory menu. You may also specify resubmission parameters that will control how frequently the Replenishment Validator polls for records in the interface tables.

Inserting into the Replenishment Interface Tables

This section provides a chart for each interface table that lists all columns, followed by a section giving a brief description of a subset of columns requiring further explanation. The chart identifies each column's datatype and whether it is Required, Derived, or Optional.

Several of the attributes in the interface tables can be populated using either the user-friendly values or the internal identifiers. For example, you have the choice of specifying either the flexfield segment representation or the internal identifier (e.g. INVENTORY_ITEM_ID) for the required value. When specifying the organization, you may

either use the organization code or the internal identifier (e.g. ORGANIZATION_ID).

If you populate the user friendly values, the Replenishment Validator will validate them and will derive the internal identifiers. If the translation is available to the external system, it may be advantageous to use the internal identifiers to improve performance.

Replenishment Headers Interface Tables

The following graphic describes the MTL_REPLENISH_HEADERS_INT table:

MTL_REPLENISH_HEADERS_INT				
Column Name	Type	Required	Derived	Optional
REPLENISHMENT_HEADER_ID	Number	✓		
REPLENISHMENT_COUNT_NAME	Varchar2(10)	✓		
COUNT_DATE	Date	✓		
LAST_UPDATE_DATE	Date	✓		
CREATION_DATE	Date	✓		
CREATED_BY	Number	✓		
LAST_UPDATE_LOGIN	Number			✓
LAST_UPDATED_BY	Number	✓		
ORGANIZATION_ID	Number	✓		
ORGANIZATION_CODE	Varchar2(3)	✓		
SUBINVENTORY_CODE	Varchar2(10)	✓		
SUPPLY_CUTOFF_DATE	Date			✓
PROCESS_STATUS	Number	✓		
PROCESS_MODE	Number	✓		
ERROR_FLAG	Number		✓	
REQUEST_ID	Number		✓	
PROGRAM_APPLICATION_ID	Number		✓	

Table 5 - 11 Oracle Inventory Replenishment Headers Interface Table (Page 1 of 2)

MTL_REPLENISH_HEADERS_INT				
Column Name	Type	Required	Derived	Optional
PROGRAM_ID	Number		✓	
PROGRAM_UPDATE_DATE	Date		✓	
DELIVERY_LOCATION_ID	Number			✓
DELIVERY_LOCATION_CODE	Varchar2(20)			✓

Table 5 – 11 Oracle Inventory Replenishment Headers Interface Table (Page 2 of 2)

ERROR_FLAG

DERIVED

If a validation error occurs, the replenishment validator populates this column with an error code. The error flag for a replenishment header will be set if either the validation of the header fails or if the validation of any of the lines of the header fails.

ORGANIZATION_ID

CONDITIONALLY REQUIRED

This column identifies the internal identifier of the organization from which the replenishment count originated. You must enter either the internal organization identifier or the user friendly organization code.

ORGANIZATION_CODE

CONDITIONALLY REQUIRED

This column is the user friendly code for the organization that is the source of the replenishment count. It may be used instead of the internal identifier, in which case the internal identifier will be derived.

PROCESS_MODE

REQUIRED

This column determines how the interfaced replenishment count will be processed. The valid options are:

2 – Concurrent

3 – Background

Interface replenishment counts marked for 'Background' processing will be picked up by the replenishment validator polling process. The validator will pick up and process all replenishment counts with a process mode of 'Background' each time it runs.

You use 'Concurrent' processing mode if you want to launch a dedicated replenishment validator process to explicitly process a single replenishment count, identified as a parameter to the program, from the interface table.

<u>PROCESS_STATUS</u>	<u>REQUIRED</u>
-----------------------	-----------------

This column is used to identify the current processing status of the replenishment count. You should insert rows that you intend to be processed with a value of 2 (Pending). The valid values for this column are:

- 1 – Hold
- 2 – Pending
- 3 – Processing
- 4 – Error
- 5 – Completed

If you want to insert records into the interface tables but temporarily prevent them from being processed, you can use this column by setting the value to 1 (Hold).

After the validator has run, it will set the value of this column to 5 (Completed). This status is used whenever the process completes, whether validation errors were detected or not.

A status of 4 (Error) indicates an internal error occurred. This error indicates an exceptional condition and should not occur.

<u>REPLENISH_HEADER_ID</u>	<u>REQUIRED</u>
----------------------------	-----------------

Enter a unique identifier for the replenishment count. This column is used to group the lines of a replenishment count with the header. You may use the sequence MTL_REPLENISH_HEADERS_S to obtain a unique identifier.

<u>REPLENISH_COUNT_NAME</u>	<u>REQUIRED</u>
-----------------------------	-----------------

Enter a unique name for the replenishment count.

<u>SUBINVENTORY_CODE</u>	<u>REQUIRED</u>
--------------------------	-----------------

This column identifies the subinventory that is the source of the replenishment count.

SUPPLY_CUTOFF_DATE**OPTIONAL**

Enter the date after which planned supply will not be considered in available quantity calculations. A null value here indicates that you do not want to consider planned supply when performing replenishment calculations.

DELIVERY_LOCATION_ID**OPTIONAL**

Enter the internal identifier for the location to which the replenishment should be delivered. You may enter the delivery location identifier, the user friendly delivery location code or neither. If neither is specified, the default delivery location for the organization from which the replenishment originated is defaulted.

DELIVERY_LOCATION_CODE**OPTIONAL**

Enter the user friendly code for the delivery location of the replenishment. You may enter this code instead of the internal identifier, in which case the internal identifier will be derived. You may specify neither the code or the identifier, in which case the default delivery location of the organization originating the replenishment will be used.

The following graphic describes the MTL_REPLENISH_LINES_INT table:

MTL_REPLENISH_LINES_INT				
Column Name	Type	Required	Derived	Optional
REPLENISHMENT_HEADER_ID	Number	✓		
REPLENISHMENT_LINE_ID	Number	✓		
ORGANIZATION_ID	Number			✓
LAST_UPDATE_DATE	Date	✓		
CREATION_DATE	Date	✓		
CREATED_BY	Number	✓		
LAST_UPDATE_LOGIN	Number	✓		
LAST_UPDATED_BY	Number	✓		
INVENTORY_ITEM_ID	Number	✓		

Table 5 – 12 Oracle Inventory Replenishment Lines Interface Table (Page 1 of 2)

MTL_REPLENISH_LINES_INT				
Column Name	Type	Required	Derived	Optional
SEGMENT {1-20}	Varchar2(40)	✓		
COUNT_TYPE_CODE	Number	✓		
COUNT_QUANTITY	Number	✓		
REFERENCE	Varchar2(240)			✓
ERROR_FLAG	Number		✓	
REQUEST_ID	Number		✓	
PROGRAM_APPLICATION_ID	Number		✓	
PROGRAM_ID	Number		✓	
PROGRAM_UPDATE_DATE	Date		✓	
COUNT_UNIT_OF_MEASURE	Varchar2(25)	✓		
COUNT_UOM_CODE	Varchar2(3)	✓		

Table 5 – 12 Oracle Inventory Replenishment Lines Interface Table (Page 2 of 2)

REPLENISHMENT_HEADER_ID

REQUIRED

Enter the unique identifier of the replenishment count. The identifier entered here is the foreign key reference which links the header table with the lines table to associate a group of lines with a single header.

REPLENISHMENT_LINE_ID

REQUIRED

Enter the identifier for the line within the replenishment count. You may use the sequence MTL_REPLENISH_LINES_S to obtain a unique identifier for the line.

INVENTORY_ITEM_ID

CONDITIONALLY REQUIRED

Enter the internal identifier for the item to be replenished.

SEGMENT{1-20}

CONDITIONALLY REQUIRED

You may use these flexfield columns instead of INVENTORY_ITEM_ID to enter the item identifier in a more user-friendly form.

ORGANIZATION_ID**OPTIONAL**

This column identifies the internal identifier of the organization from which the replenishment count originated. If you do not enter a value here, the organization identifier will be derived from the replenishment header.

COUNT_TYPE_CODE**REQUIRED**

Enter the type of the replenishment count entry. The valid count types are:

1 – On-hand Quantity

2 – Order Quantity

3 – Order Maximum

Use 'On-hand quantity' to identify counts that are the result of stock-takes of subinventories in which you do not track on-hand quantities.

Use 'Order quantity' when you want to specify the quantity to be ordered. This count type may be used with either tracked or non-tracked subinventories.

Use 'Order maximum' when you want to place an order for the min-max maximum quantity specified for item in the subinventory specified. This count type may be used with either tracked or non-tracked subinventories.

COUNT_QUANTITY**CONDITIONALLY REQUIRED**

This column is used to specify the count quantity that corresponds to the count type entered for the line. When the count type is 'On-hand quantity', the count quantity is the on-hand balance determined during the stock-take. When the count type is 'Order quantity', the count quantity represents the quantity to be ordered. This column is not used when the count type is 'Order maximum'.

REFERENCE**OPTIONAL**

Use this column to enter any replenishment count reference information.

COUNT_UNIT_OF_MEASURE **CONDITIONALLY REQUIRED**

Enter the count unit of measure identifier. This column may be used to specify the full name for the unit of measure. This column is meaningful only when a value is entered in the COUNT_QUANTITY columns.

COUNT_UOM_CODE **CONDITIONALLY REQUIRED**

This column represents the unit of measure code used for the count. You may specify the code when populating this table or you may use the full name for the unit of measure, in which case this column will be derived. This column is meaningful only when a value is entered in the COUNT_QUANTITY columns.

ERROR_FLAG **DERIVED**

This flag indicates the error status of the validation of a replenishment line. The replenishment validator populates this column with a line corresponding to the error detected during validation.

Validation

Oracle Inventory validates the following conditions:

- The value of REPLENISH_HEADER_ID must be unique among existing replenishment counts
- The value of REPLENISH_COUNT_NAME must be unique among existing count headers
- The value of LAST_UPDATED_BY must be a valid user name
- ORGANIZATION_ID must be a valid identifier of an organization
- SUBINVENTORY_CODE must refer to an existing subinventory
- DELIVERY_LOCATION_ID must be a valid identifier of a location associated with the organization generating the replenishment.
- There must be at least one line per header
- The ORGANIZATION_ID at the header level must be the same as that at the line level
- COUNT_TYPE_CODE must be either "1" or "2" or "3" and must be consistent with whether the subinventory is tracked or non-tracked

- The value of COUNT_QUANTITY must be consistent with COUNT_TYPE_CODE and must be greater than zero
- INVENTORY_ITEM_ID must refer to a transactable item in the organization specified
- The item must exist in the subinventory and must be min-max planned in that subinventory
- The COUNT_UOM_CODE must be valid and conversions to primary UOM must exist
- Each line must correspond to a header

Viewing Failed Transactions

Replenishment counts that fail the validation process will remain in the MTL_REPLENISH_HEADERS_INT and MTL_REPLENISH_LINES_INT tables. You may use SQL*PLUS to identify the headers that have failed by selecting those rows with a process_status of 5 (Complete). The reason for the failure will be reflected in the ERROR_FLAG column.

Possible values for the ERROR_FLAG column in the MTL_REPLENISH_HEADERS_INT table are:

- 1 – Non-unique replenishment header id
- 2 – Non-unique replenishment count name
- 3 – Invalid user name
- 4 – Invalid organization identifier
- 5 – Invalid subinventory
- 7 – Header with no corresponding replenishment lines
- 10 – Header failed because line failed
- 18 – Delivery location is not valid

Possible values for the ERROR_FLAG column in the MTL_REPLENISH_LINES_INT table are:

- 1 – No corresponding header id
- 3 – Invalid user name
- 8 – Invalid item identifier or item isn't transactable
- 9 – Invalid unit of measure or no conversion to primary unit of measure exists

- 11 – No item specified in either identifier or segments
- 12 – Invalid count type
- 13 – On-hand count type used for tracked subinventory
- 14 – Invalid count quantity
- 15 – Lines organization header does not match header organization identifier
- 17 – Item is not specified in the subinventory or is not min-max planned in the subinventory

Fixing Failed Transactions

Frequently, errors in the interface are caused by problems external to the replenishment count itself. For example, there may be validation that failed because an entity that was being validated had the wrong status (i.e. disabled), or the failure could even be the result of a system error, such as running out of space. In these cases, the resolution is simple; once you have made the necessary changes, you simply need to resubmit replenishment validator process.

If, however, you need to make changes to the data in the interface table, you need to either delete the failed records, correct them in the external feeder system and resubmit them, or update the interface record in the interface table using SQL*PLUS. When you resubmit updated transactions for processing, all validation will be performed again.

Open Item Interface

You can import items from any source into Oracle Inventory and Oracle Engineering using the Item Interface. With this interface, you can 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 import new items from a Product Data Management package. The Item Interface validates your data, insuring that your imported items contain the same item detail as items you enter manually in Oracle Inventory's Define Item form or Oracle Engineering's Define Engineering Item form.



Defining and Maintaining Items
(*Oracle Inventory Reference Manual*)

The purpose of this essay is to explain how to use the Item Interface.

Functional Overview

The Item Interface lets you import items into Oracle Inventory and, if installed at your site, Oracle Engineering. When you import items through the Item Interface, you create new items in your item master organization or assign existing items to additional organizations. You can specify values for all the item attributes, or you can specify just a few attributes and let the remainder default or remain null. The Item Interface also lets you import revision details, including past and future revisions and effectivity dates. Validation of imported items is done using the same rules as the item definition forms, so you are insured of valid items.



Define Item
(*Oracle Inventory Reference Manual*)



Define Engineering Item
(*Oracle Engineering Reference Manual*)

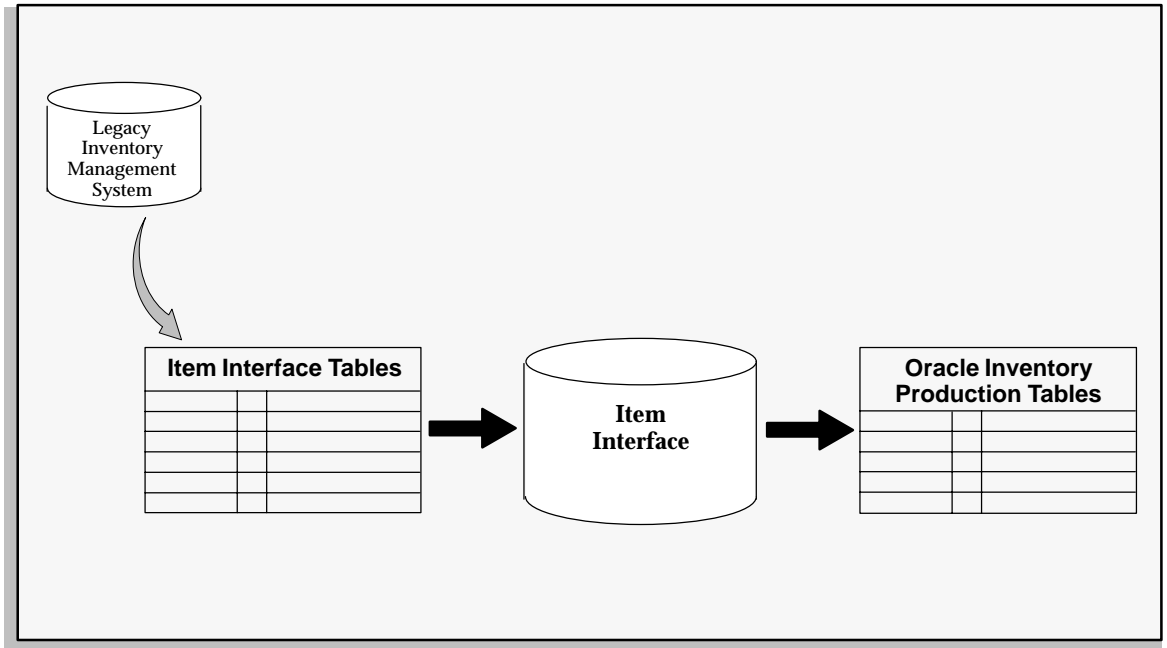


Engineering Prototype Environment
(*Oracle Engineering Reference Manual*)


The Item Interface reads data from two tables for importing items and item details. You use the MTL_SYSTEMS_ITEM_INTERFACE table for your new item numbers and all item attributes. This is the main item interface table, and may be the only table you choose to use. If you are

importing revision details for your new items, you can use the `MTL_ITEM_REVISIONS_INTERFACE` table. This table is used only for revision information, and is not required. A third table, `MTL_INTERFACE_ERRORS`, is used for error tracking of all items that the Item Interface fails.

Figure 5 – 4



Before you use the Item Interface, you must write and run a custom program that extracts item information from your source system and inserts it into the `MTL_SYSTEM_ITEM_INTERFACE` table, and (if revision detail is included) the `MTL_ITEMS_REVISIONS_INTERFACE` table. After you load the items into these interface tables, you run the Item Interface to import the data. The Item Interface assigns defaults, validates data you include, and then imports the new items.

 **Attention:** You must import items into a master organization before you import items into additional organizations. You can accomplish this by specifying only your master organization on a first pass run of the Item Interface. Once this has completed, you can run the Item Interface again, this time specifying an additional or all organizations.

You can also use the Item Interface to import item material cost, material overhead, and revision details.

Setting Up the Item Interface

Add the Item Interface to the Menu

Before you use the Item Interface, add it to the Oracle Inventory menu system, so that it can be run from the menu and launched as a concurrent program. To add the Item Interface to the menu, log onto Oracle Applications and choose the Application Developer responsibility. Navigate to the Define Menu form. Choose an appropriate menu for the Item Interface, such as Enter Item Information. In the Menu Entries portion of the screen, specify a sequence number as well as a prompt and a description for the Item Interface. Enter the prompt (such as **Interface**) and description (such as **Item Interface**) just as you want them to appear on your menu. Under the Action section, set type to **Form**, Application to **Application Object Library**, Name to **Run Reports**, and Arguments to **GLOBAL.REQUEST_GROUP_APPL_SHORT_NAME="INV" GLOBAL.REQUEST_GROUP_CODE="INV_ITEM_IMPORT" GLOBAL.TITLE="Process Interface Items"**.



Define Menu

(Oracle Applications System Administration Reference Manual)

Create Indexes for Performance

You should create the following indexes to improve Item Interface performance.

First, determine which segments are enabled for the System Items flexfield.

Then, for example, if you have a two-segment flexfield, with segment8 and segment12 enabled, you would do the following:

```
SQL> create unique index MTL_SYSTEM_ITEMS_UC1 on mtl_system_items
(organization_id, segment8, segment12);
SQL> create unique index MTL_SYSTEM_ITEMS_INTERFACE_UC1 on
mtl_system_items_interface (organization_id, segment8, segment12);
```

If you plan to populate the ITEM_NUMBER column in mtl_system_items_interface instead of the item segment columns, do not create the MTL_SYSTEM_ITEMS_INTERFACE_UC1 unique index. Instead, create MTL_SYSTEM_ITEMS_INTERFACE_NC1 non-unique index on the same columns.

Start the Concurrent Manager

Since you launch and manage the Item Interface concurrent program through the concurrent manager, you must ensure that the concurrent manager is running before you can import any items.

Set Profile Option Defaults



Attention: Some columns use profile options as default values. You must set these profiles if you want them to default.



Profile Options in Oracle Inventory
(*Oracle Inventory Reference Manual*)



Defining and Maintaining Items
(*Oracle Inventory Reference Manual*)



Setting Up Oracle Inventory
(*Oracle Financials and Oracle Government Financials Implementation Manual*)

Item Interface Runtime Options

When you run the Item Interface, you are prompted for report parameters. These are runtime options for the Item Interface. These options include:

All Organizations

Yes	Run the interface for all organization codes in the item interface table.
No	Run the interface only for the organization you are currently in. Item interface rows for organizations other than your current organization are ignored.

Validate Items

Yes	Validate all items and their data residing in the interface table that have not yet been validated. If items are not validated, they will not be processed into Oracle Inventory.
No	Do not validate items in the interface table. Note that items that have not been validated will not be

processed into Oracle Inventory. You would use this option if you had previously run the item interface and responded **Yes** for **Validate Items** and **No** for **Process Items**, and now want to process your items.

Process Items

Yes	All qualifying items in the interface table are inserted into Oracle Inventory.
No	Do not insert items into Oracle Inventory. Use this option, along with Yes for Delete Processed Items , to remove successfully processed rows from the interface table without performing any other processing. You can also use this option, along with Yes for Validate Items , if you want to validate items without any processing.

Delete Processed Rows

Yes	Delete successfully processed items from the item interface tables.
No	Leave all rows in the item interface tables.

Process Set

Enter a number for the set id for the set of rows you want to process. The program will pick up the rows marked with that id in the SET_PROCESS_ID column. If you leave this field blank, all rows are picked up for processing regardless of the SET_PROCESS_ID column value.

Inserting into the Item Interface Table

Item Interface Table Description

The item interface table MTL_SYSTEM_ITEMS_INTERFACE contains every column in the Oracle Inventory item master table, MTL_SYSTEM_ITEMS. The columns in the item interface correspond directly to those in the item master table. Except for ITEM_NUMBER or SEGMENTn columns, ORGANIZATION_CODE or ORGANIZATION_ID, and DESCRIPTION, all of these columns are

optional, either because they have defaults that can be derived, or because the corresponding attributes are optional and may be left null.



Attention: The item interface table contains additional columns, some of which appear in Table 5 – 13. Some columns in the item interface table are *not* used by the interface; these columns are *not* listed in Table 5 – 13. Data entered in those columns is ignored.

The item costing columns (those that begin **MATERIAL_...**) and the **REVISION** column *are* used for importing item costs and revisions and are discussed in a later section of this chapter.

You may also put in details about other interface tables not used by the Item Open Interface.

As currently running, the interface does not support the
MTL_CROSS_REFERENCE_INTERFACE,
MTL_ITEM_CATEGORIES_INTERFACE, or
MTL_SECONDARY_LOCS_INTERFACE.

The **MTL_ITEM_CATEGORIES_INTERFACE** is used by the **ITEM_OPEN_INTERFACE** internally, but should not be populated by the user.

MTL_SYSTEM_ITEMS_INTERFACE (Partial List of Columns) Column Name				
	Type	Required	Derived	Optional
ITEM_NUMBER	Varchar2(81)	<i>conditionally</i>		
DESCRIPTION	Varchar2(240)	<i>conditionally</i>		
MATERIAL_COST	Number			✓
MATERIAL_OVERHEAD_RATE	Number			✓
MATERIAL_OVERHEAD_SUB_ELEM	Varchar2(50)			✓
MATERIAL_OVERHEAD_SUB_ELEM_ID	Number			✓
MATERIAL_SUB_ELEM	Varchar2(50)			✓
MATERIAL_SUB_ELEM_ID	Number			✓
ORGANIZATION_CODE	Varchar2(3)	<i>conditionally</i>		
PROCESS_FLAG	Number	✓		
REVISION	Varchar2(3)			✓

Table 5 – 13 Partial List of Columns, Oracle Inventory Item Interface Table (Page 1 of 2)

MTL_SYSTEM_ITEMS_INTERFACE (Partial List of Columns) Column Name				
	Type	Required	Derived	Optional
TRANSACTION_ID	Number		✓	
TRANSACTION_TYPE	Varchar2(5)	✓		
SET_PROCESS_ID	Number	✓		

Table 5 – 13 Partial List of Columns, Oracle Inventory Item Interface Table (Page 2 of 2)

Required Data


Every row in the item interface table must identify the item and organization. To identify the item when importing it, you may specify either the ITEM_NUMBER or SEGMENT n columns—the Item Interface generates the INVENTORY_ITEM_ID for you. Specifying either the ORGANIZATION_ID or ORGANIZATION_CODE adequately identifies the organization. When more than one of these columns has been entered and they conflict, ITEM_NUMBER overrides SEGMENT n and ORGANIZATION_ID overrides ORGANIZATION_CODE. It is strongly recommended that you use SEGMENT column instead of ITEM_NUMBER.




Warning: If you enter a value for the ITEM_NUMBER column and you are using a multi-segment item, you must insert the system item flexfield separator between each segment of your item number. For example, if you are using a two segment item and have defined a dash (-) as your separator, a typical item would be entered as 1234-5678. When the item interface program derives the item's segment values, it searches for this separator to indicate the end of one segment value and the start of the next segment value. In our example, 1234 would be put in SEGMENT1, 5678 in SEGMENT2.



Warning: If you enter values for SEGMENT n columns, be sure that the segments you use correspond to the key flexfield segments you defined for your items. No validation for the correct segments occurs when you run the Item Interface. Also, the open item interface program expects that all segments that you use for the system item flexfield be required segments. Your system items flexfield should not be defined with any optional segments.

 **Attention:** No segment validation is done against value sets.

 Define Key Flexfield Segments
(*Oracle Applications Flexfields Manual*)

When you import a new item, you are also required to specify the DESCRIPTION. This is not required when assigning an existing or already imported item to an additional organization, as the description can be defaulted down from the master record. (Of course, if the description is at the item–organization level, you are always able to override the master organization description by giving this column a value.)

There are two other columns the Item Interface uses to manage processing. They are TRANSACTION_TYPE, which tells the Item Interface how to handle the row, and PROCESS_FLAG, which indicates the current status of the row.

Always set the TRANSACTION_TYPE column to **CREATE**, to create an item record (true when both importing a new item and assigning an already existing item to another organization). This is the only value currently supported by the Item Interface.

The Item Interface uses the PROCESS_FLAG to indicate whether processing of the row succeeded or failed. When a row is ready to be processed, give the PROCESS_FLAG a value of 1 (Pending), so that the Item Interface can pick up the row and process it into the production tables.

Code	Meaning
1	Pending
2	Assign complete
3	Assign/validation failed
4	Validation succeeded; import failed
5	Import in process
7	Import succeeded

Table 5 – 14 Meaning of PROCESS_FLAG Values (Page 1 of 1)

A full list of values for the PROCESS_FLAG is in Table 5 – 14, but you are unlikely to see all of these.

Other columns, although required in the production tables, are not required in the item interface table, because they have default values or

their values can be derived from other sources. Check the defaults and derived values carefully, as they may not be the values you desire.

If the Item Interface successfully processes a row in the item interface table or the revision interface table, the program sets the PROCESS_FLAG to 7 (Import succeeded) for the row. If the Item Interface cannot insert a row into the production table, the PROCESS_FLAG column for the failed row is set to 4 (Import failed). If a row in the interface table fails validation, the PROCESS_FLAG column is set to 3 (validation failed). A row is inserted into the MTL_INTERFACE_ERRORS table for all failed rows. You can review and update any failed rows in each interface table using custom reports and programs.

Derived Data

Many columns have defaults that the Item Interface uses when you leave that column null in the item interface table. Columns with defaults are listed in Table 5 – 15.

MTL_SYSTEM_ITEMS_INTERFACE (Partial List of Columns) Column Name			Default Value	Value Displayed in form
SUMMARY_FLAG ¹			Y	
ENABLED_FLAG			Y	
PURCHASING_ITEM_FLAG			N	No
SHIPPABLE_ITEM_FLAG			N	No
CUSTOMER_ORDER_FLAG			N	No
INTERNAL_ORDER_FLAG			N	No
SERVICE_ITEM_FLAG			N	No
SERVICE_STARTING_DELAY_DAYS			0	0
INVENTORY_ITEM_FLAG			N	No
ENG_ITEM_FLAG ²			N	No
INVENTORY_ASSET_FLAG			N	No
PURCHASING_ENABLED_FLAG			N	No

Table 5 – 15 Column Defaults in the Item Interface Table (Page 1 of 5)

MTL_SYSTEM_ITEMS_INTERFACE (Partial List of Columns) Column Name			Default Value	Value Displayed in form
CUSTOMER_ORDER_ENABLED_FLAG			N	No
INTERNAL_ORDER_ENABLED_FLAG			N	No
SO_TRANSACTIONS_FLAG			N	No
MTL_TRANSACTIONS_ENABLED_FLAG			N	No
STOCK_ENABLED_FLAG			N	No
BOM_ENABLED_FLAG			N	No
BUILD_IN_WIP_FLAG			N	No
WIP_SUPPLY_TYPE			1	Push
REVISION_QTY_CONTROL_CODE			1	Not under revision quantity control
ALLOW_ITEM_DESC_UPDATE_FLAG			<i>from PO_SYSTEM_PARAMETERS_ALL. ALLOW_ITEM_DESC_UPDATE_FLAG</i>	<i>from Define Purchasing Options, otherwise Yes</i>
RECEIPT_REQUIRED_FLAG			<i>from PO_SYSTEM_PARAMETERS_ALL. RECEIVING_FLAG</i>	<i>from Define Purchasing Options, otherwise No</i>
RFQ_REQUIRED_FLAG			<i>from PO_SYSTEM_PARAMETERS_ALL. RFQ_REQUIRED_FLAG</i>	<i>from Define Purchasing Options, otherwise No</i>
LOT_CONTROL_CODE			1	No lot control
SHELF_LIFE_CODE			1	No shelf life control
SERIAL_NUMBER_CONTROL_CODE			1	No serial number control
RESTRICT_SUBINVENTORIES_CODE			2	Subinventories not restricted to predefined list
RESTRICT_LOCATORS_CODE			2	Locators not restricted to predefined list
LOCATION_CONTROL_CODE			1	No locator control
PLANNING_TIME_FENCE_CODE			4	User-defined time fence
PLANNING_TIME_FENCE_DAYS			1	1
BOM_ITEM_TYPE			4	Standard
PICK_COMPONENTS_FLAG			N	No
REPLENISH_TO_ORDER_FLAG			N	No

Table 5 – 15 Column Defaults in the Item Interface Table (Page 2 of 5)

MTL_SYSTEM_ITEMS_INTERFACE (Partial List of Columns) Column Name			Default Value	Value Displayed in form
ATP_COMPONENTS_FLAG			N	No
ATP_FLAG			N	No
PRIMARY_UNIT_OF_MEASURE			<i>from profile</i> INV: Default Primary Unit of Measure	<i>from</i> Update Personal Profile Options
ALLOWED_UNITS_LOOKUP_CODE			3	Both standard and item specific
COST_OF_SALES_ACCOUNT			<i>from</i> MTL_PARAMETERS. COST_OF_SALES_ACCOUNT	<i>from</i> Define Organization Parameters
SALES_ACCOUNT_DSP			<i>from</i> MTL_PARAMETERS.SALES_ACCOUNT	<i>from</i> Define Organization Parameters
ENCUMBRANCE_ACCOUNT			<i>from</i> MTL_PARAMETERS.ENCUMBRANCE_ACCOUNT	<i>from</i> Define Organization Parameters
EXPENSE_ACCOUNT			<i>from</i> MTL_PARAMETERS.EXPENSE_ACCOUNT	<i>from</i> Define Organization Parameters
LIST_PRICE_PER_UNIT			0	0
INVENTORY_ITEM_STATUS_CODE			<i>from profile</i> INV: Default Item Status	<i>from</i> Update Personal Profile Options
INVENTORY_PLANNING_CODE			6	Not planned
PLANNING_MAKE_BUY_CODE			2	Buy
MRP_SAFETY_STOCK_CODE			1	Non-MRP planned
TAXABLE_FLAG			Y	<i>from</i> Define Purchasing Options, otherwise No
MATERIAL_BILLABLE_FLAG ³			M	Material
EXPENSE_BILLABLE_FLAG ⁴			N	No
TIME_BILLABLE_FLAG ⁴			N	No
SERVICE_DURATION			0	0
MARKET_PRICE			0	0
PRICE_TOLERANCE_PERCENT			0	0

Table 5 – 15 Column Defaults in the Item Interface Table (Page 3 of 5)

MTL_SYSTEM_ITEMS_INTERFACE (Partial List of Columns) Column Name			Default Value	Value Displayed in form
SHELF_LIFE_DAYS			0	0
RESERVABLE_TYPE			1	Reservable
REPETITIVE_PLANNING_FLAG			N	No
ACCEPTABLE_RATE_DECREASE			0	0
ACCEPTABLE_RATE_INCREASE			0	0
END_ASSEMBLY_PEGGING_FLAG			N	None
POSTPROCESSING_LEAD_TIME			0	0
VENDOR_WARRANTY_FLAG			N	No
SERVICEABLE_COMPONENT_FLAG			N	No
SERVICEABLE_PRODUCT_FLAG			Y	Yes
PREVENTIVE_MAINTENANCE_FLAG			N	No
SHIP_MODEL_COMPLETE			N	No
RETURN_INSPECTION_REQUIREMENT			2	Inspection not required
PRORATE_SERVICE_FLAG			N	No
INVOICEABLE_ITEM_FLAG			N	No
INVOICE_ENABLED_FLAG			N	No
MUST_USE_APPROVED_VENDOR_FLAG			N	No
OUTSIDE_OPERATION_FLAG			N	No
COSTING_ENABLED_FLAG			N	No
CYCLE_COUNT_ENABLED_FLAG			N	No
AUTO_CREATED_CONFIG_FLAG			N	No
MRP_PLANNING_CODE			6	Not planned
CONTAINER_ITEM_FLAG			N	No
VEHICLE_ITEM_FLAG			N	No
END_ASSEMBLY_PEGGING_FLAG ⁵			N	None

Table 5 – 15 Column Defaults in the Item Interface Table (Page 4 of 5)

MTL_SYSTEM_ITEMS_INTERFACE (Partial List of Columns) Column Name		
	Default Value	Value Displayed in form
SERVICE_DURATION ⁵	0	0
SET_PROCESS_ID	0	

Table 5 – 15 Column Defaults in the Item Interface Table (Page 5 of 5)

Notes:

¹Defaulted to Y by the Item Open Interface, but the item form defaults N for this column.

²Defaulted to N by the Item Open Interface, but in the item form the default value depends on whether the form is accessed from Oracle Engineering.

³For use only at Release 10.7 or later. New values M, E, and L will be allowed from a patch release of 10.7. Currently only Y and N are allowed.

⁴At Release 10.7 and forward, this is obsolete but is defaulted to N by the Item Open Interface for backward compatibility. The item form defaults null. A future patch release of 10.7 may default null for these.

⁵Will be in a patch release of 10.7.

You can import item descriptive flexfield values when you have implemented a descriptive flexfield for items. To do this, simply include values for the descriptive flexfield columns (ATTRIBUTE_CATEGORY and ATTRIBUTE n columns) in the item interface table. No validation is performed on descriptive flexfield values.

In addition, the Item Interface uses the item's status (INVENTORY_ITEM_STATUS_CODE) to determine the value of attributes under status control. If an attribute is under status control, then the attribute value always derives from the item's status, and any value in the attribute column of the item interface table is ignored. If an attribute is under default status control, then the attribute value only derives from the item's status if there is no value in the attribute column of the item interface table. If an attribute is not under any status control, then the item status has no effect on the attribute's value for the imported item.



Warning: If an attribute is under status control, it still must follow the attribute dependency rules. For example, if the

BOM_ENABLED_FLAG is under status control, and a status is used setting BOM_ENABLED_FLAG to Yes, the INVENTORY_ITEM_FLAG must be set to **Yes** for the imported item. If the item has INVENTORY_ITEM_FLAG set to **No** (or it is left null and therefore defaults to **No**), the Item Interface processes the item with the BOM_ENABLED_FLAG set to **No**. This is because the attribute dependency rules stipulate that BOM_ENABLED_FLAG can only be **Yes** for an Inventory Item.



Attention: When you assign an item to a child organization, all item-level attributes default down from the master organization—but only when the attribute column is *null* in the item interface table. If you supply a value for a item-level attribute in a child organization record, the Item Interface rejects the record as an error. The exception is status attributes under status control. These attributes *always* derive from the item's status, never from the master record. See Table 5 – 15 for the list of defaults supplied by the Item Interface.

Whether you import a new item to an master organization or assign an existing item to a non-master organization, the Item Interface always enters a unique numeric identifier in the TRANSACTION_ID column of the item interface table, and the concurrent request number in the REQUEST_ID column of the item master table.

Item Categories

When the Item Interface imports an item, it also assigns the item to the mandatory category sets based on the item defining attributes. The default category for each category set is used. The Item Interface does not allow you to assign items to other category sets, nor does the interface allow you to specify an item's category value.



Define Category Set
Define Default Category Set
(Oracle Inventory Reference Manual)

For example, suppose you define a category set *Inventory* with a default category of *glass*, and you designate *Inventory* as the mandatory category set for inventory items. When the interface imports an inventory item (INVENTORY_ITEM_FLAG = Y), the item is assigned to the *glass* category in the *Inventory* category set, and a corresponding row is inserted into MTL_ITEM_CATEGORIES.

When using the Item Interface to assign an existing item to another organization, the item is also assigned to mandatory category sets with the default category. As described above, the item defining attributes

determine to which mandatory category sets the item is assigned. Even if the item is assigned to a item level category set (non-mandatory) in the master organization, it is not assigned to that category set in the item's new organization.

Validation

When you import an item, the Item Interface validates the data and any derived values the same way manually entered items are validated. This validation ensures that:

- required columns have an included or defaulted value
- control levels are reflected in item attribute values
- status control settings for status attributes are maintained
- interdependences between item attribute values are consistent



Warning: Before you can import an item into a child organization, it must exist in the master organization. You cannot import an item into both a master organization and a child organization at the same time. If you populate the item interface table with both master and child item records, you should run the Item Interface for the master organization only. After it has successfully finished running, run the interface again for all organizations. This second run inserts the items into the child organizations.



Defining and Maintaining Items
(*Oracle Inventory Reference Manual*)



Define Item
Define Item Attribute Controls
(*Oracle Inventory Reference Manual*)



Define Engineering Item
(*Oracle Engineering Reference Manual*)

When you import items, the Item Interface program validates all rows in the table that have a PROCESS_FLAG set to **1** (Pending). The interface first assigns the default values to the derived columns of the row, then updates the value of the PROCESS_FLAG column to **2** (Assign Succeeded).

The Item Interface then validates each row. If a row in the interface table fails validation, the program sets the PROCESS_FLAG to **3**

(Assign/Validation Failed) and inserts a row into the table explaining the error.

For all successfully validated rows, the interface inserts the rows into Oracle Inventory's item master table, MTL_SYSTEM_ITEMS. If a row cannot be inserted into the item master table, the program sets the PROCESS_FLAG to 4 (Import Failed) and inserts a row into the errors table.

After this program inserts the imported item into the item master table, the row is deleted or the PROCESS_FLAG is set to 7 (Import Succeeded).

To minimize the number of rows stored in the interface table, you can specify at run time that the program delete successfully processed records after insertion. If you do not delete successfully processed records automatically, you can write custom programs that report and delete any successfully imported rows. The program can search for rows with a PROCESS_FLAG value of 7 (Import Succeeded), list the rows in a report, and then delete them from the table. By defining a report set in Oracle Application Object Library, you can automatically run the custom program after each submission of the Item Interface. You can also run multiple Item Open Interface processes. See: Multi-Thread Capability below.

Importing Additional Item Details

You can import additional cost and revision details for an imported item using interface tables listed in Table 5 – 16. The Item Interface program imports the details specified in these tables at the same time that it imports the items themselves. The program validates all rows you insert into the interface tables, derives additional columns, and creates the item and item details in Oracle Inventory.

Item Detail	Interface Table	Number of Rows per Item
Costs	MTL_SYSTEM_ITEMS_INTERFACE	1
Revisions	MTL_SYSTEM_ITEMS_INTERFACE (for imported items only)	1
	MTL_ITEM_REVISIONS_INTERFACE	1 or more

Table 5 – 16 Oracle Inventory Item Details Interface Tables



Warning: Although there are many other tables in Oracle Inventory whose names may imply use, the tables listed in Table 5 – 16 are the *only* interface tables used by the open Item Interface in Release 10.5 to import item details.

Before importing additional item details, you must complete the same setup steps required for manually defining these item details. For example, you must define your cost types and activities before you can assign item costs. The default cost category set must be specified using the Define Default Category Sets form, and the starting Revision must be set for all organizations.



Defining and Maintaining Items
(*Oracle Inventory Reference Manual*)



Setting Up Oracle Inventory
(*Oracle Financials and Oracle Government Financials Implementation Manual*)

The Item Interface validates all required data and some optional data included in the item detail interface tables. When you import your cost or revision data, this program validates the included or derived values the same way Oracle Inventory validates manually entered details.

Importing Cost Details

When the Item Interface imports an item, it may also import costing information into Oracle Cost Management tables. The interface may import this costing information automatically using organization and category defaults, or you may specify the information for the item itself in the item interface table.

If you set up a default material overhead rate for the item's organization or for the default cost category, this material overhead rate is inserted into the cost details table, CST_ITEM_COST_DETAILS, and summarized in the item costs table, CST_ITEM_COSTS.



Define Material Overhead Defaults
(*Oracle Inventory Reference Manual*)

You may specify one material cost and one material overhead rate for the item, directly in the item interface table itself. Remember to include the material sub-element for the material cost and overhead rate by specifying the sub-element.

Importing Revision Details

You can import detailed revision history with your new items in any one of the following ways:

- specify revisions and effectivity dates in the revision interface table
- specify the current revision for each item in the item interface table
- do not specify any revisions and let the Item Interface default the revision based on the starting revision defined in the Define Organization Parameters form

To import multiple item revisions and effectivity dates, use the revision interface table, `MTL_ITEM_REVISIONS_INTERFACE`. You may also include ECN information (see Table 5 – 17). You need to create your own program for populating this table.

Since revisions exist at the item–organization level, you need revision data for each item–organization you are updating. Include a row for each revision (with an effectivity date) to import, in ascending order. In other words, for each item–organization combination, revision A must have an effectivity date less than or equal to revision B, and so on. Each row in this table must correspond to a row in the item interface table. Each row must reference the item’s `ITEM_NUMBER` and `ORGANIZATION_ID` or `ORGANIZATION_CODE`.



Attention: When importing multiple revisions for the same item, if one of the revisions fails validation, all revisions for that item fail.

To import an item and its current revision only, include a value for the `REVISION` column in the item interface table. The Item Interface automatically creates this revision with an effective date equal to the system date when it imports the item. (Use the revision interface table described above if you wish to specify a revision effectivity date.)

If you choose not to use the revision interface table, and do not include a revision in the item interface table, the Item Interface assigns each item a beginning revision, using the default specified in each Organization Parameter. The system date is the effectivity date. Once established, you cannot add revisions with effectivity dates earlier than the date assigned by the Item Interface.



Warning: Although most item information defaults from the master organization when you assign an existing item to a child

organization, the Item Interface does *not* default an item's revision detail from the master organization.



Define Item Revisions (Oracle Inventory Reference Manual)

As with the item interface table, the column PROCESS_FLAG indicates the current state of processing for a row in the revision interface table. Possible values for the column are listed in Table 5 – 14.

When you insert rows into the revision interface table, you should set the PROCESS_FLAG to 1 (Pending) and TRANSACTION_TYPE to CREATE.

MTL_ITEM_REVISIONS Column Name	MTL_ITEM_REVISIONS_INTERFACE Column Source
INVENTORY_ITEM_ID	ITEM_NUMBER
ORGANIZATION_ID	ORGANIZATION_ID or ORGANIZATION_CODE
REVISION	REVISION
CHANGE_NOTICE	CHANGE_NOTICE
ECN_INITIATION_DATE	ECN_INITIATION_DATE
IMPLEMENTATION_DATE	IMPLEMENTATION_DATE
IMPLEMENTED_SERIAL_NUMBER	IMPLEMENTED_SERIAL_NUMBER
EFFECTIVITY_DATE	EFFECTIVITY_DATE
ATTRIBUTE_CATEGORY	ATTRIBUTE_CATEGORY
ATTRIBUTE _n	ATTRIBUTE _n
REVISED_ITEM_SEQUENCE_ID	REVISED_ITEM_SEQUENCE_ID
DESCRIPTION	DESCRIPTION

Table 5 – 17 Column–Mappings from Revision Interface Table to Oracle Inventory

Additional Information:

Oracle Inventory Technical Reference Manual

You can import revision descriptive flexfield values when you have implemented a descriptive flexfield for revisions. To do this, simply include values for the descriptive flexfield columns (ATTRIBUTE_CATEGORY and ATTRIBUTE_n columns) in the revision interface table when you import revisions.

The Item Interface may also be used to create revisions for existing items. Only revision labels and effectivity dates higher than existing revisions may be imported. To do this, simply load revision detail for the existing items into MTL_ITEM_REVISIONS_INTERFACE and run the Concurrent Item Interface program.

Resolving Failed Interface Rows

If a row fails validation, the Concurrent Item Interface program sets the PROCESS_FLAG to 3 (Assign/validation failed) and inserts a row in the interface errors table, MTL_INTERFACE_ERRORS. To identify the error message for the failed row, the program automatically populates the TRANSACTION_ID column in this table with the TRANSACTION_ID value from the corresponding item interface table. For example, if a row in the item interface table fails, the program inserts a row into the interface errors table with the failed row's TRANSACTION_ID as the error row's TRANSACTION_ID. Each error in the interface errors table has a value for the MESSAGE_NAME and REQUEST_ID columns. The Item Interface populates these columns for item detail errors the same way it populates the table for item errors.

The UNIQUE_ID column in MTL_INTERFACE_ERRORS is populated from the sequence MTL_SYSTEM_ITEMS_INTERFACE_S. Thus, for a given row, the sequence of errors can be determined by examining UNIQUE_ID for a given TRANSACTION_ID.

For example, if your row with TRANSACTION_ID 2000 failed with two errors in the MTL_INTERFACE_ERRORS table, then you can see which error occurred first by looking at the row with the smallest UNIQUE_ID for TRANSACTION_ID 2000.

You should resolve errors in the sequence that they were found by the interface program, that is, in increasing order of UNIQUE_ID for any TRANSACTION_ID.

Quite often, resolving the first few errors and restarting the Item Open Interface will cause the other (spurious) errors for that failed row to disappear.

The Concurrent Item Interface program automatically inserts validated rows into the production tables in Oracle Inventory and Oracle Cost Management. Depending on the item information you import, the Item Interface inserts these rows into the item master, item categories, item costs, cost details, and item revisions tables. If a row cannot be inserted into one of these tables, the PROCESS_FLAG column for all

remaining rows is set to 4 (Import failed) and the Concurrent Item Interface inserts a row in the interface errors table. The program handles these processing errors in the same way it handles validation errors.

Reviewing Failed Rows

You can review and report rows in any of the interface tables using SQL*Plus or any custom reports you develop. Since all rows in the interface tables have a value for PROCESS_FLAG, you can easily identify records that have not been successfully imported into Oracle Inventory.

Correcting Failed Rows

Use SQL*Plus to delete and re-add rows from the interface tables. If there are any changes to be made in the failed row, you should delete the row, add the original row with the required changes, and then process.

Multi-Thread Capability (Parallel Runs of the Item Open Interface)

In Release 10.7 and above, the following tables will have a new NOT NULL NUMBER column called SET_PROCESS_ID:

- MTL_SYSTEM_ITEMS_INTERFACE
- MTL_ITEM_REVISIONS_INTERFACE
- MTL_ITEM_CATEGORIES_INTERFACE

The SET_PROCESS_ID column has a database default value of zero in the three tables above.

To have parallel runs of the Item Open Interface, the SET_PROCESS_ID column for records in the interface tables has to be populated with a positive, nonzero number.

Example:

You have 1000 records in the MTL_SYSTEM_ITEMS_INTERFACE table that you want to insert into MTL_SYSTEM_ITEMS, and you decide to have four parallel Item Open Interface processes to accomplish this task.

In the scripts you use to insert data into the MTL_SYSTEM_ITEMS_INTERFACE table, populate the first 250

records with SET_PROCESS_ID = 1, the next 250 records with SET_PROCESS_ID = 2, and so on.



Attention: If you have custom scripts to insert data into the MTL_SYSTEM_ITEMS_INTERFACE table, you must modify them to include the SET_PROCESS_ID column. Also remember that any corresponding records that you enter in the MTL_ITEM_REVISIONS_INTERFACE table should have the matching SET_PROCESS_ID values.

From the Item Open Interface SRS launch form, specify the Process Set for a given run in the Process Set parameter. This will initiate four Item Open Interface concurrent programs with the Process Set parameter set to 1, 2, 3, and 4 respectively. Four Item Open Interface processes will run in parallel, each working on the set you specified.



Attention: Leaving a null in the Process Set parameter will process all rows regardless of the process set value. This is similar to the Item Open Interface behavior before this patch. If you do not want the new multi-thread capability, you can populate the interface tables as you always have. The SET_PROCESS_ID column will get the default value of zero. When you run the Item Open Interface with the Process Set parameter blank, the interface processes all rows (regardless of the SET_PROCESS_ID value) as it did in earlier releases.

Multi-threading Rules

The Applications DBA for the site should enforce these rules:

- If you run an Item Open Interface process with the Process Set value null, you should not concurrently run any other Item Open Interface processes.
- Do not have parallel runs of the Item Open Interface on the same process set.

Not following these rules will cause multiple Item Open Interface processes trying to work on the same set of rows and will lead to unpredictable errors.

Oracle Order Entry Open Interfaces

This chapter contains information about the following Oracle Order Entry open interfaces:

- Ship Confirm Open Interface
- Integrating Oracle Order Entry with Oracle Receivables
- Integrating Oracle Order Entry Using OrderImport

Ship Confirm Open Interface

The Ship Confirm Open Interface provides a way to load externally derived Shipping data into picking tables and close the pick slip without using the Confirm Shipments window. The Ship Confirm Open Interface takes data loaded into three interface tables and:

- validates the information contained within the interface tables,
- loads the valid data into the picking header, picking line details, and freight charges tables, and
- closes the pick slip (if requested).

If you close the pick slip using the Ship Confirm Open Interface and the OE: Immediate Inventory Update profile option is set to No, you can run the Update Shipping and Inventory Interface programs to update order lines with shipped quantities and to update inventory. If the OE: Immediate Inventory Update profile option is set to Yes, the Update Shipping and Inventory Interface programs run automatically when you close the pick slip.

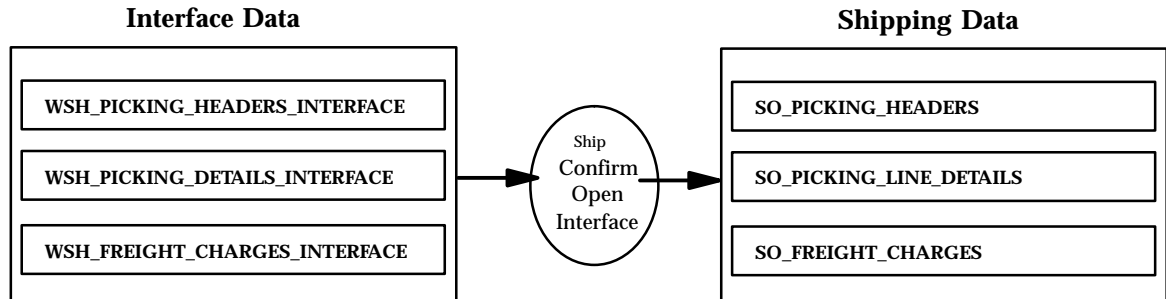
Functional Overview

The basic workflow of the Ship Confirm Open Interface consists of:

- entering data in the picking header, picking line details, and freight charges interface tables via an external device (such as a bar code reader), and
- running the Ship Confirm Open Interface concurrent process from the Standard Report Submission window.



Attention: If you want to run the Update Shipping Information and Inventory Interface concurrent programs online, you must start the Shipping Transaction Manager (see Shipping Transaction Manager: page 6 – 13).



The Ship Confirm Open Interface reads data from the interface tables, performs all necessary validation on the data, loads the data into Shipping tables, and executes a close action (if requested). Pick slip data can be divided between the three interface tables, but one record for the pick slip must be input into the WSH_PICKING_HEADERS_INTERFACE table. Picking line detail and freight charges tables may have many records (or no records at all). The complete set of records must be grouped by a common TRANSACTION_ID. Each transaction is processed for a single pick slip, but a pick slip can have multiple transactions. Once the pick slip is closed, however, no further transactions for the pick slip will be processed.

Transactions

A transaction consists of data belonging to a pick slip that may have picking line details or freight charge information. A transaction can contain one or more of the following actions:

- update picking header (waybill number, picker, packer, date shipped, expected arrival date, number of boxes, weight, and unit of measure)
- update picking line details (shipped quantities and inventory controls)
- add freight charges
- close pick slip with either a backorder or ship-confirm status

For each record in the WSH_PICKING_HEADERS_INTERFACE table, the Ship Confirm Open Interface processes the information in the following order:

1. If picking line details exist in the WSH_PICKING_DETAILS_INTERFACE view for the transaction, the Ship Confirm Open Interface updates the SO_PICKING_LINE_DETAILS table.
2. If freight charges exist in the WSH_FREIGHT_CHARGES_INTERFACE table for the transaction, the Ship Confirm Open Interface inserts the freight charges into the SO_FREIGHT_CHARGES table.
3. If any of the key fields in the WSH_PICKING_HEADERS_INTERFACE table are not null, the Ship Confirm Open Interface updates the SO_PICKING_HEADERS table.
4. If the ACTION_CODE column in the WSH_PICKING_HEADERS_INTERFACE table is set to CLOSE-BACKORDER, the Ship Confirm Open Interface closes the pick slip for the transaction and backorders any items that are not defined as shipped. If ACTION_CODE is set to CLOSE-SHIP, the Ship Confirm Open Interface sets the shipped quantity equal to the released quantity where a shipped quantity was not defined, closes the pick slip, and backorders any items that are not defined as shipped.

The primary key on the WSH_PICKING_HEADERS_INTERFACE table is the TRANSACTION_ID, which is generated using the MTL_MATERIAL_TRANSACTIONS_S sequence. The TRANSACTION_ID is also the foreign key to both the WSH_PICKING_DETAILS_INTERFACE and the WSH_FREIGHT_CHARGES_INTERFACE tables.

When uploading freight charges and picking line data using the Ship Confirm Open Interface, the following rules apply:

- Once inventory controls are updated, they cannot be changed.
- Shipped quantity can only be increased, not decreased.
- Freight charges can only be inserted, not updated.

For example, say you processed a transaction that updated the picking details and defined a shipped quantity of 2. If you discovered an error and tried to correct it by resubmitting the transaction with modified data, the following would result:

- If you tried to change the inventory controls, the Ship Confirm Open Interface interprets the new transaction as a separate ship confirmation and inserts a new record for the new inventory controls with the same shipped quantity.
- If you tried to correct the shipped quantity by defining a lower value of 1, the Ship Confirm Open Interface interprets this as a separate ship confirmation and increments the shipped quantity by 1 (total shipped quantity now equals 3).

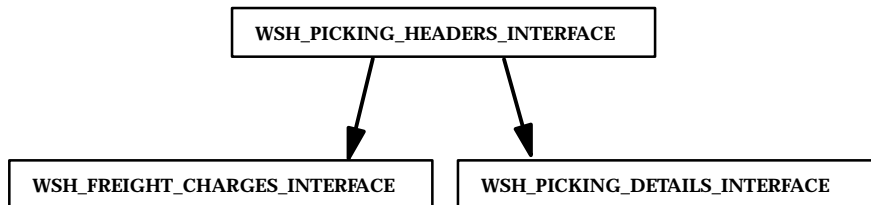
If you want to correct inventory control information or make changes to the quantity shipped, you must use the Confirm Shipments window. If you want to update freight charge information, you must use the Freight Charges window.

Inserting into the Ship Confirm Open Interface Tables

This section provides a chart for each interface table that lists all input columns, followed by a section giving a brief description of a subset of columns requiring further explanation. The chart identifies each column's datatype and whether it is Required (R), Derived (D), Optional (O), or Not Applicable (NA) for the function. These tables are all available to the APPS user_id.

Picking Headers Interface Table

The WSH_PICKING_HEADERS_INTERFACE table drives the Ship Confirm Open Interface.



The following table describes the WSH_PICKING_HEADERS_INTERFACE table:

Column Name	Type	Update Header	Update Detail	Close Slip	Update Freight
TRANSACTION_ID	Number	R	R	R	R
PROCESS_FLAG	Number	R	R	R	R
PICK_SLIP_NUMBER	Number	R	R	R	R
ACTION_CODE	Varchar2(15)	NA	NA	R	NA
CREATION_DATE	Date	R	R	R	R
CREATED_BY	Number	R	R	R	R
LAST_UPDATE_DATE	Date	D	D	D	D
LAST_UPDATED_BY	Number	D	D	D	D
LAST_UPDATE_LOGIN	Number	D	D	D	D
DOCUMENT_SET	Varchar2(30)	NA	NA	O	NA
DOCUMENT_SET_ID	Number	NA	NA	O	NA
WAYBILL_NUM	Varchar2(50)	O	NA	R*	NA
DATE_SHIPPED	Date	O	NA	O	NA
CONTAINERS	Number	O	NA	O	NA
EXPECTED_ARRIVAL_DATE	Date	O	NA	O	NA
PICKED_BY_ID	Number	O	NA	O	NA
PICKED_BY_NAME	Varchar2(30)	O	NA	O	NA
PACKED_BY_ID	Varchar2(30)	O	NA	O	NA
PACKED_BY_NAME	Number	O	NA	O	NA
FREIGHT_CARRIER_CODE	Varchar2(30)	O	NA	O	NA
UOM_CODE	Varchar2(3)	O	NA	O	NA
UNIT_OF_MEASURE	Varchar2(25)	O	NA	O	NA
WEIGHT	Number	O	NA	O	NA
NUMBER_OF_BOXES	Number	O	NA	O	NA

Table 6 – 1 Oracle Order Entry / Shipping Picking Headers Interface Table (Page 1 of 1.

* -- Required if any items are shipped. Not required if items are completely backordered.

PROCESS_FLAG

This column indicates the processing status. The valid options are:

1. To Be Processed
2. Processed
3. Error

You must set this column to 1 to enable processing. If an error occurs while processing, this column is set to 3. You must correct the error and reset this column to 1 to process the transaction again. An error of 3 prevents all downstream processing for the pick slip even if other transactions have their PROCESS_FLAG set to 1.

ACTION_CODE

This column defines whether and how you want to close the pick slip. The valid options are 1 and 2. All other values are ignored.

- 1 (Close-complete) — closes the pick slip, ships those items with a defined Shipped Quantity, and completely ships any items on the pick slip that do not have defined shipped quantities.
- 2 (Close-partial) — closes the pick slip and ships only those items with a defined shipped quantity.

DOCUMENT_SET / DOCUMENT_SET_ID

These columns define the document set that you want printed when the pick slip is closed. You may input either the internal id or the text name. If you input both, the DOCUMENT_SET_ID will be used.

WAYBILL_NUM

This column defines the waybill number. You must define a waybill number when you close a pick slip containing shipped items. This column is not required when you close a pick slip if all items on the pick slip are backordered.

DATE_SHIPPED

This column defines the date that the pick slip is shipped. This column defaults to SYSDATE if it is not defined when you ship the pick slip.

EXPECTED_ARRIVAL_DATE

This column defines the arrival date of the shipment. This column is optional and defaults to the DATE_SHIPPED if not specified.

FREIGHT_CARRIER_CODE

Defines the Freight Carrier Code. Defaults to the freight carrier specified at order entry (SO_HEADERS_SHIP_METHOD_CODE).

UOM_CODE / UNIT_OF_MEASURE

Defines the unit of measure. This column is required when you close the pick slip.

Picking Line Details Interface

The WSH_PICKING_DETAILS_INTERFACE is a *view* that physically maps onto the MTL_TRANSACTIONS_INTERFACE (MTI) table. This view represents all pick slip details information. The PROCESS_FLAG column must be set to 9. The following table describes the WSH_PICKING_DETAILS_INTERFACE view:

Column Name	Null?	Type	Notes
PROCESS_FLAG	Not Null	Number(1)	Must be set to 9
TRANSACTION_ID	Not Null	Number	
PICKING_LINE_DETAIL_ID	Not Null	Number	
SOURCE_CODE	Not Null	Varchar2(30)	
SOURCE_HEADER_ID	Not Null	Number	
TRANSACTION_MODE	Not Null	Number	
TRANSACTION_TYPE_ID	Not Null	Number	
LAST_UPDATE_DATE	Not Null	Date	
LAST_UPDATED_BY	Not Null	Number	
CREATION_DATE	Not Null	Date	
CREATED_BY	Not Null	Number	

Table 6 – 2 Oracle Order Entry / Shipping Picking Details Interface Table (Page 1 of 3)

Column Name	Null?	Type	Notes
LAST_UPDATE_LOGIN		Number	
ERROR_EXPLANATION		Varchar2(240)	output only
ERROR_CODE		Varchar2(240)	output only
INVENTORY_ITEM_ID		Number	
INVENTORY_ITEM		Varchar2(2000)	
WAREHOUSE_ID	Not Null	Number	
SUBINVENTORY		Varchar2(10)	
LOT_NUMBER		Varchar2(30)	
REVISION		Varchar2(3)	
LOCATOR_ID		Number	
LOCATOR_NAME		Varchar2(2000)	
SERIAL_NUMBER		Varchar2(30)	
SHIPPED_QUANTITY	Not Null	Number	
TRANSACTION_UOM	Not Null	Varchar2(3)	
TRANSACTION_DATE	Not Null	Date	
ATTRIBUTE_CATEGORY		Varchar2(30)	DFF for table
ATTRIBUTE1		Varchar2(150)	
ATTRIBUTE2		Varchar2(150)	
ATTRIBUTE3		Varchar2(150)	
ATTRIBUTE4		Varchar2(150)	
ATTRIBUTE5		Varchar2(150)	
ATTRIBUTE6		Varchar2(150)	
ATTRIBUTE7		Varchar2(150)	
ATTRIBUTE8		Varchar2(150)	
ATTRIBUTE9		Varchar2(150)	
ATTRIBUTE10		Varchar2(150)	

Table 6 – 2 Oracle Order Entry / Shipping Picking Details Interface Table (Page 2 of 3)

Column Name	Null?	Type	Notes
ATTRIBUTE11		Varchar2(150)	
ATTRIBUTE12		Varchar2(150)	
ATTRIBUTE13		Varchar2(150)	
ATTRIBUTE14		Varchar2(150)	
ATTRIBUTE15		Varchar2(150)	

Table 6 – 2 Oracle Order Entry / Shipping Picking Details Interface Table (Page 3 of 3)

PICKING_LINE_DETAIL_ID

This column is input from the Pick Slip report. Use the same PICKING_LINE_DETAIL_ID for each inventory control configuration used to meet this picking line. For example, if a picking line is split between 2 subinventories, input one record for each of the subinventories. Each record must reference the original PICKING_LINE_DETAIL_ID.

PROCESS_FLAG

This column must be set to 9. Any value other than 9 may cause critical errors in the Inventory Transaction Manager.

INVENTORY_ITEM_ID / INVENTORY_ITEM

This required column must be the same as defined on the picking line.

WAREHOUSE_ID

This required column must be the same as defined on the picking line.

SUBINVENTORY

If a reservation has been placed, this column cannot be modified from the picking line. If no reservation has been placed, this column is required for validation. If null is input, the item's default subinventory is used for this column if defined.

LOT_NUMBER

If a reservation has been placed, this column cannot be modified. If no reservation has been placed, this column is only valid and required if the item is under lot control.

REVISION

If a reservation has been placed, this column cannot be modified. If no reservation has been placed, this column is only valid and required if the item is under reservation control.

LOCATOR_ID / LOCATOR_NAME

If a reservation has been placed, this column cannot be modified. If no reservation has been placed, this column is only valid and required if the item is under location control.

SERIAL_NUMBER

This column is only valid and required if the item is under serial number control.

Freight Charges Interface

This table loads freight charges. The following table describes the WSH_FREIGHT_CHARGES_INTERFACE table:

Column Name	Null?	Type	Notes
TRANSACTION_ID	Not Null	Number	System generated unique number
CREATION_DATE	Not Null	Date	
CREATED_BY	Not Null	Number	
LAST_UPDATE_DATE	Not Null	Date	
LAST_UPDATED_BY	Not Null	Number	
LAST_UPDATE_LOGIN	Not Null	Number	
FREIGHT_CHARGE_TYPE_ID		Number	
FREIGHT_CHARGE_TYPE_DESC		Varchar2(30)	

Table 6 - 3 Oracle Order Entry / Shipping Freight Charges Interface Table (Page 1 of 2)

Column Name	Null?	Type	Notes
AMOUNT	Not Null	Number	
CURRENCY_CODE		Varchar	
CURRENCY_NAME		Varchar	

Table 6 – 3 Oracle Order Entry / Shipping Freight Charges Interface Table (Page 2 of 2)

Validation

Data inserted into the interface tables must be validated and any relational integrity must be checked. The validation performed ensures that data is identical to data generated by the Confirm Shipments window. Oracle Order Entry / Shipping validates the following conditions:

- The value of PICKED_BY_NAME / PICKED_BY_ID must be valid in FND_USER.
- The value of PACKED_BY_NAME / PACKED_BY_ID must be valid in FND_USER.
- DOCUMENT_SET / DOCUMENT_SET_ID must be valid in SO_REPORT_SETS.
- UNIT_OF_MEASURE / UOM_CODE must be valid in MTL_UNITS_OF_MEASURE.
- WAREHOUSE_ID must be valid in ORG_ORGANIZATION_DEFINITIONS.
- FREIGHT_CARRIER_CODE must be valid in ORG_FREIGHT for the organization.
- FREIGHT_CHARGE_TYPE_ID / FREIGHT_CHARGE_TYPE_DESC must be valid in SO_FREIGHT_CHARGE_TYPES as FREIGHT_CHARGE_TYPE_ID/NAME.
- CURRENCY_CODE / CURRENCY_NAME must be either freight currency or order currency.
- EXPECTED_ARRIVAL_DATE must be greater than or equal to the DATE_SHIPPED if specified.

- Subinventory must be valid for the item and not disabled. Lot must be valid for the item, organization, and subinventory. Serial number must be valid for the item, organization, revision, or lot when using predefined serial number control. Serial numbers must be unique. Locator must be valid for the item and not disabled.
- The shipped quantity cannot exceed the requested quantity.
- The sum of the shipped quantities of the picking line details cannot exceed the shipped quantity of the parent picking line.

Viewing Failed Transactions

Transactions that fail the validation process are labeled with a `PROCESS_FLAG=3` on the header record (`WSH_PICKING_HEADERS_INTERFACE`). The `ERROR_CODE` column will be populated with a description of the error on the record which caused the error. If the error did not occur on the header record, the `ERROR_CODE` column will indicate which of the two detail tables caused the error.

Fixing Failed Transactions

If you encounter failed transactions, you can delete the failed records, correct the errors in the external feeder system and resubmit them, or update the interface record in the interface table using `SQL*PLUS`. When you resubmit updated transactions for processing, all validation will be performed again. When you resubmit the transaction, make sure that the `PROCESS_FLAG` column is set to 1.

Shipping Transaction Manager

Before running the Ship Confirm Open Interface, you must start the Shipping Transaction Manager. The steps involved for starting the Shipping Transaction Manager using Release 10 versions of Order Entry / Shipping are listed below.



Attention: The Internal Manager must be started for the Shipping Transaction Manager to run.

To start the Shipping Transaction Manager:

1. Change to the System Administrator responsibility.
2. Navigate to the Administer Concurrent Managers form (Navigate:Concurrent:Manager:Administer).
3. Query the Shipping Transaction Manager.
4. Navigate to the Control field.
5. Select Activate concurrent manager from the Control QuickPick.

The Target field determines the maximum number of manager processes that can be active for the Shipping Transaction Manager. You can define the Target field (the maximum number of processes) in the Define Concurrent Manager form. If the Target field is already defined (has a quantity greater than zero defined), the Shipping Transaction Manager will run automatically when the Internal Manager is started.
6. Save your work.

To define the maximum number of manager processes:

1. Change to the System Administrator responsibility.
2. Navigate to the Define Concurrent Manager form (Navigate:Concurrent:Manager:Define).
3. Query the Shipping Transaction Manager.
4. Navigate to the Concurrent Manager Details region.
5. Select Work Shifts from the Concurrent Manager Details QuickPick.
6. Navigate to the Work Shifts region.
7. Define the appropriate number of Target Processes.
8. Save your work.

Integrating Oracle Order Entry with Oracle Receivables

Oracle Order Entry provides functionality to integrate with Oracle Receivables, and using AutoInvoice, create invoices, create credit memos and credits on account, recognize revenue, and manage sales credits.

Basic Needs

Oracle Order Entry and Oracle Receivables provide features you need to satisfy the following integration needs:

- To create accurate and timely invoices, credit memos and credits on account from order entry transactions.
- To control when order transactions are invoiced.

Major Features

Receivables Interface

Oracle Order Entry provides a powerful program to automatically collect order and return information and populate the Oracle Receivables AutoInvoice interface tables. You control, using process parameters, which types of orders are interfaced each time you execute the program. Oracle Order Entry ensures that all orders or order lines and returns or return lines have successfully completed any required prerequisites.



Using the Power of Order Cycles
(*Oracle Order Entry Reference Manual*)



Run Concurrent Programs
(*Oracle Order Entry Reference Manual*)

Invoice Source

The Invoice Source QuickPick for the Receivables Interface parameter requires that an invoice source be set up with specific values for AutoInvoice Validation. When defining Invoice Sources in Oracle Receivables, you must create at least one invoice source for Order

Entry use if you want to interface orders and returns to Oracle Receivables for processing by AutoInvoice.

The following table (Table 6 – 4) shows the values that the fields of the Batch Source zone and the AutoInvoice Data Validation zone of the Define Invoice Sources form must have so that you can choose your invoice sources in the Invoice Source parameter field when you run the Receivables Interface from the Run Concurrent Programs form.

Field in Define Invoice Source form	Necessary Value
<i>Invoice Source zone</i>	
Type	Imported
Status	Active
Automatic Invoice Numbering	Yes
<i>AutoInvoice Data Validation zone</i>	
Transaction Type	Id
Sold-to Customer	Id
Bill-to Customer	Id
Bill-to Address	Id
Bill-to Contact	Id
Ship-to Customer	Id
Ship-to Address	Id
Ship-to Contact	Id
Payment Method Rule	(any)
Customer Bank Account	(any)
Invoicing Rule	Id
Accounting Rule	Id
Accounting Flexfield	(any)
Derive Date	(any)
Payment Terms	Id
Revenue Account allocation	(any)
Memo Reason	Id

Table 6 – 4 (Page 1 of 2)

Field in Define Invoice Source form	Necessary Value
<i>AutoInvoice Data Validation zone</i>	
Agreement	Id
Memo Line Rule	Id
Sales Territory	(any)
Inventory Item	Id
Unit of Measure	Id
FOB Point	Code
Freight Carrier	Code
Related Document	Id
<i>Sales Credits Data Validation zone</i>	
Salesperson	Id
Sales Credit Type	Id
Sales Credit	Percent

Table 6 – 4 (Page 2 of 2)

Automatic Tax Calculation

AutoInvoice automatically calculates sales tax as orders from Oracle Order Entry are processed. Taxing is based on the Sales Tax Location Flexfield combination. If you have designated a customer as tax exempt, AutoInvoice will not tax any items billed for the customer. Or if you have designated an item as non-taxable, AutoInvoice will not tax the item.

Automatic Account Code Creation

Oracle Receivables uses AutoAccounting to determine the revenue account for all transactions from Oracle Order Entry. AutoAccounting lets you define what information is used to define the various segments of your Accounting Flexfield.



Using AutoAccounting
(*Oracle Receivables Reference Manual*)

Accounting and Invoicing Rules

Oracle Order Entry uses accounting and invoicing rules. This information is transferred to Oracle Receivables to be used to determine the invoice date (invoicing rule) and general ledger distribution records (accounting rule). Oracle Order Entry passes an invoicing rule and accounting rule for each order transaction interfaced to Oracle Receivables, except if the accounting rule is **Immediate**, in which case Oracle Order Entry does not pass any value (inserts null).

Accounting Rules

Oracle Order Entry determines the accounting rule for sales order lines based on the following hierarchy.

Accounting Rule Hierarchy for Sales Order Lines	
1	If you referenced an agreement on the order that <i>does not</i> allow override of the accounting rule, Oracle Order Entry inserts the accounting rule from the associated agreement (SO_AGREEMENTS.ACCOUNTING_RULE_ID), if not, then...
2	If you referenced a commitment on the order line that is associated with an agreement that <i>does not</i> allow override of the accounting rule, Oracle Order Entry inserts the accounting rule from the agreement (SO_AGREEMENTS.ACCOUNTING_RULE_ID), if not, then...
3	If you defined an accounting rule for the item Oracle Order Entry will use the accounting rule for the item (MTL_SYSTEM_ITEMS.ACCOUNTING_RULE_ID), if not, then...
4	If you referenced a commitment on the order that is associated with an agreement that <i>does</i> allow override of the accounting rule, Oracle Order Entry inserts the accounting rule from the associated agreement (SO_AGREEMENTS.ACCOUNTING_RULE_ID), if not, then...
5	If you referenced an agreement on the order line that <i>does</i> allow override of the accounting rule, Oracle Order Entry inserts the accounting rule from the agreement (SO_AGREEMENTS.ACCOUNTING_RULE_ID), if not, then...
6	In all other cases, Oracle Order Entry inserts the accounting rule on the order.

Table 6 – 5 (Page 1 of 1)

Invoicing Rules

Oracle Order Entry determines the invoicing rule for a sales order line based on the following hierarchy:

Invoicing Rule Hierarchy for Sales Order Lines	
1	If you referenced an agreement on the order that <i>does not</i> allow override of the invoicing rule, Oracle Order Entry inserts the invoicing rule from the agreement (SO_AGREEMENTS.INVOICING_RULE_ID), if not, then...
2	If you referenced a commitment on the order line that is associated with an agreement that <i>does not</i> allow override of the invoicing rule, Oracle Order Entry inserts the invoicing rule from the agreement (SO_AGREEMENTS.INVOICING_RULE_ID), if not, then...
3	If you defined an invoicing rule for the item Oracle Order Entry will use the invoicing rule for the item (MTL_SYSTEM_ITEMS.INVOICING_RULE_ID), if not, then...
4	If you referenced a commitment on the order line that is associated with an agreement that <i>does</i> allow override of the invoicing rule, Oracle Order Entry inserts the invoicing rule from that agreement (SO_AGREEMENTS.INVOICING_RULE_ID), if not, then...
5	If you referenced an agreement on the order that <i>does</i> allow override of the invoicing rule, Oracle Order Entry inserts the invoicing rule from the agreement. (SO_AGREEMENTS.INVOICING_RULE_ID), if not, then...
6	In all other cases, Oracle Order Entry inserts the invoicing rule on the order.

Table 6 – 6 (Page 1 of 1)



Define Invoicing and Accounting Rules
Enter Commitments
(*Oracle Receivables Reference Manual*)

Define Item
(*Oracle Inventory Reference Manual*)

Define Agreements
(*Oracle Order Entry Reference Manual*)

Credit Method for Accounting Rule

Oracle Order Entry transfers a Credit Method for Accounting Rule for each return line. This credit method is recognized only by invoices that

use *duration* accounting rules. You can assign a Credit Method for Accounting Rule to the order type of the return. If the Credit Method for Accounting Rule field for the order type is null, then Oracle Order Entry transfers **LIFO** (Last In First Out).



Understanding Credit Memos
(*Oracle Receivables Reference Manual*)



Enter Credit Memos
(*Oracle Receivables Reference Manual*)

Credit Method for Installments

Oracle Order Entry transfers a Credit Method for Installments for each return line. This credit method is used for crediting an invoice that uses split payment terms. You can assign a Credit Method for Installments to the order type of the return. If the Credit Method for Installments field for the order type is null, then Oracle Order Entry transfers **LIFO** (Last In First Out).

Internal Sales Orders

Oracle Order Entry does not process internal sales order lines for the Receivables Interface, even if the Receivables Interface is an action in the order cycle for the internal sales order.

Internal sales orders are orders that originate in Oracle Purchasing as internal requisitions, and are imported to Oracle Order Entry as internal sales orders using OrderImport.

Invoicing of ATO Configurations

Invoicing Attributes

For ATO configurations, Oracle Order Entry considers the base model's item attribute of a configuration to see if it should consider passing invoice information to Oracle Receivables for each order line in the configuration. If you have the item attributes **Invoiceable Item** and **Invoice Enabled** set to Yes for the base model item, Oracle Order Entry then considers these item attributes for each component in the bill of material for the model to see if they should be invoiced in Oracle Receivables. If the item attributes **Invoiceable Item** or **Invoice Enabled** are set to No for the base model item, Oracle Order Entry does not pass invoicing information to Oracle Receivables for any order

lines for the components within the configuration, regardless of the item attribute settings.



Item Attributes Appendix
(*Oracle Order Entry Reference Manual*)

Required for Revenue Attribute

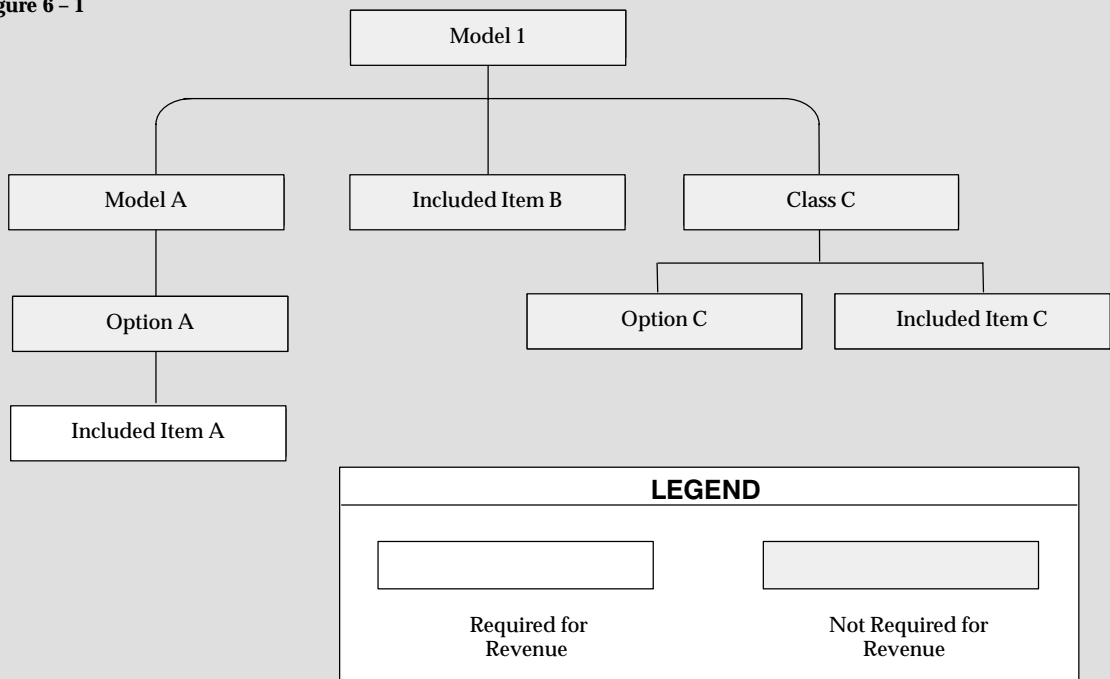
The bill of material attribute, **Required for Revenue**, allows you to define specific items in a bill that must be shipped before their parent can be invoiced. In all cases the control applies to only one level, the immediate parent. Except for classes, the control relationship is the child affecting the parent. The following diagrams demonstrate some examples.



Bills of Material
(*Oracle Bills of Material Reference Manual*)

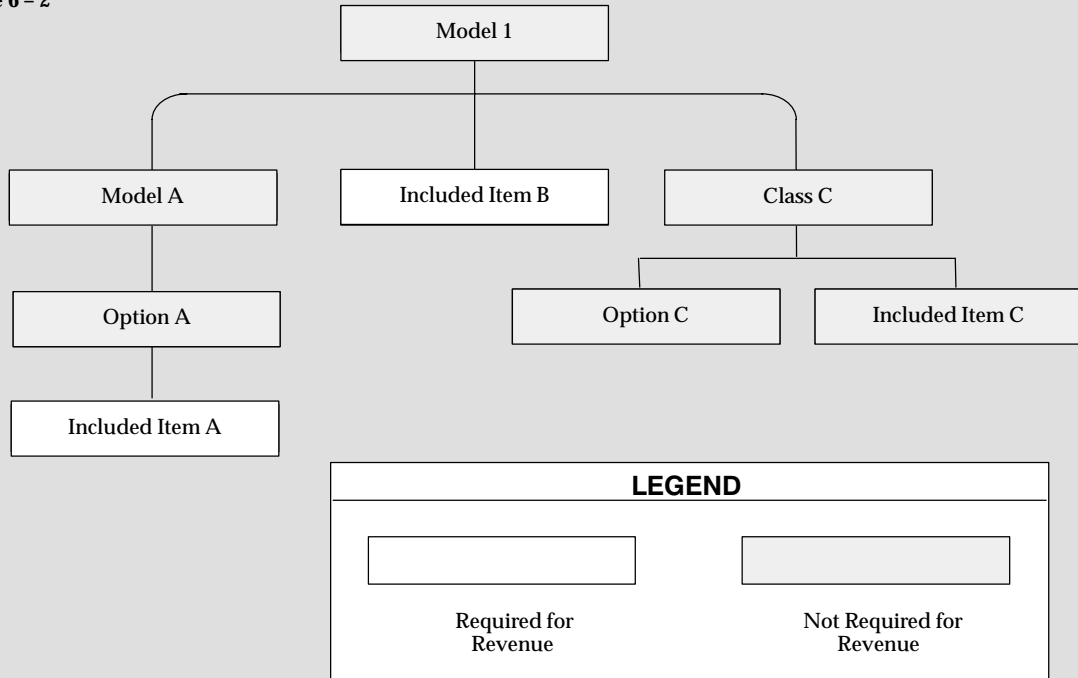
Item Attributes Appendix
(*Oracle Order Entry Reference Manual*)

Figure 6 – 1



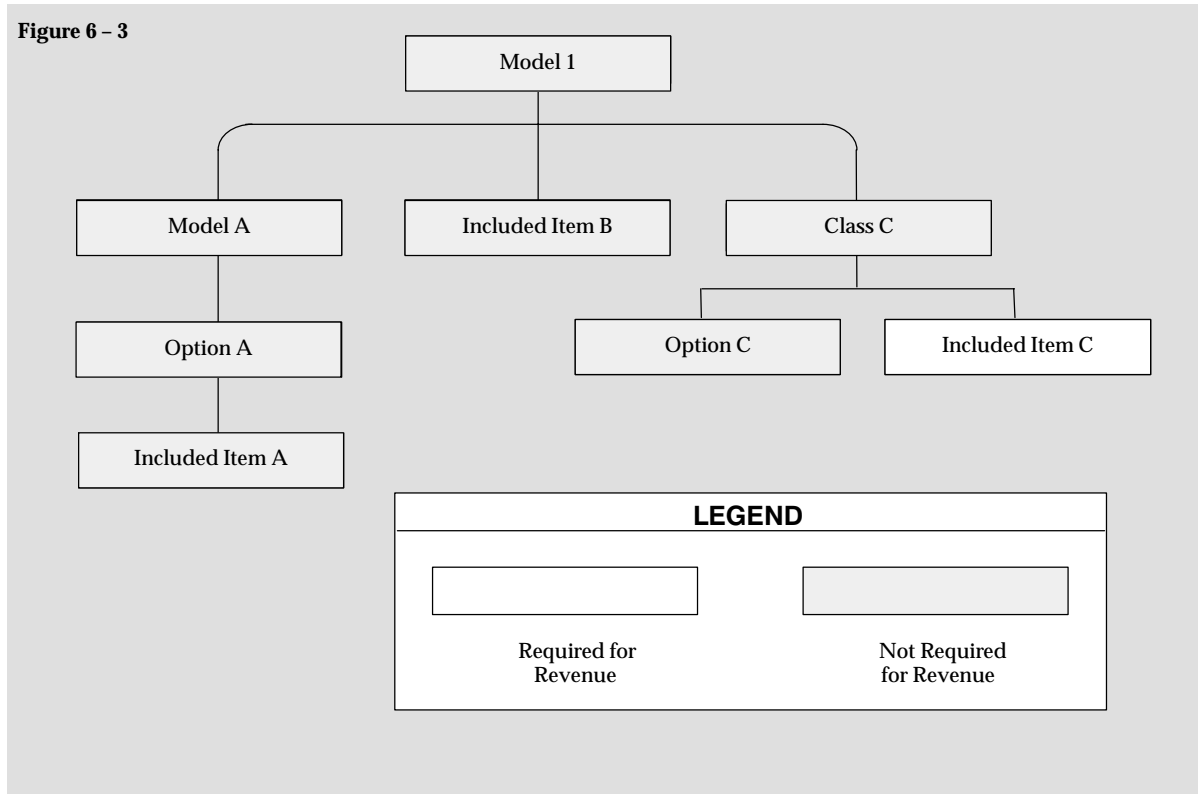
In Figure 6 – 1, Included Item A has the Required for Revenue attribute set to Yes. Option A is not eligible to interface to Oracle Receivables until Included Item A is shipped, even if Option A is also shippable and has shipped. All other components, including Model A and Model 1, are eligible to interface regardless of Included Item A's shipment status.

Figure 6 – 2



In Figure 6 – 2, Included Item B has the Required for Revenue attribute set to Yes. Model 1 is not eligible to interface to Oracle Receivables until Included Item B is shipped. And again, Option A is not eligible to interface until Included Item A is shipped.

Figure 6 – 3



The situation with classes is unique. If any item below a class in a bill has the Required for Revenue attribute set to Yes, then that item must be shipped before the parent item *and* the other items in the class are eligible to interface. For example, in Figure 6 – 3, Included Item C has the Required for Revenue attribute set to Yes. Therefore, both Option C and Class C are not eligible to interface until Included Item C is shipped.

Understanding the Interface Tables

Oracle Order Entry inserts information into two of the three AutoInvoice interface tables (RA_INTERFACE_LINES and RA_INTERFACE_SALES_CREDITS). RA_INTERFACE_DISTRIBUTIONS is not described in this essay because all account code creation is done by AutoInvoice based on the AutoAccounting rules you have defined. The following describes what information

Oracle Order Entry interfaces for each order and order line, each sales credit, and each freight charge.

RA_INTERFACE_LINES

INTERFACE_LINE_ID	NUMBER(15)
-------------------	------------

Oracle Order Entry does not insert a value into this column.

INTERFACE_LINE_CONTEXT	VARCHAR2(30)
------------------------	--------------

Oracle Order Entry inserts the value you have set for the profile option, **OE: Source Code**.

INTERFACE_LINE_ATTRIBUTE1	VARCHAR2(30)
---------------------------	--------------

Oracle Order Entry inserts SO_HEADERS. ORDER_NUMBER.

INTERFACE_LINE_ATTRIBUTE2	VARCHAR2(30)
---------------------------	--------------

Oracle Order Entry inserts SO_ORDER_TYPES.NAME.

INTERFACE_LINE_ATTRIBUTE3	VARCHAR2(30)
---------------------------	--------------

Shipped order line	Oracle Order Entry inserts SUBSTR(SO_PICKING_HEADERS.PICK_SLIP_NUMBER, 1, 30); this is the first thirty characters of a fifty character field.
---------------------------	--

Non-Shipped order line	Oracle Order Entry inserts 0 (zero).
-------------------------------	--------------------------------------

Return lines	Oracle Order Entry inserts 0 (zero).
---------------------	--------------------------------------

Freight charges	Oracle Order Entry inserts SO_PICKING_HEADERS.PICK_SLIP_NUMBER.
------------------------	---

INTERFACE_LINE_ATTRIBUTE4	VARCHAR2(30)
---------------------------	--------------

Shipped order line	Oracle Order Entry inserts SUBSTR(SO_PICKING_HEADERS.WAYBILL_NUMBER, 1, 30).
---------------------------	--

Non-Shipped order line	Oracle Order Entry inserts 0 (zero).
Return lines	Oracle Order Entry inserts 0 (zero).
Freight charges	Oracle Order Entry inserts SUBSTR(SO_PICKING_HEADERS.WAYBILL_NUMBER, 1, 30).
	<u>INTERFACE_LINE_ATTRIBUTE5</u> <u>VARCHAR2(30)</u>
Sales Order lines and Return lines	Oracle Order Entry inserts the number of times the order or return line has been interfaced for invoice or credit (SO_LINES.INVOICE_COUNT+1).
Freight charges	Oracle Order Entry inserts 1.
	<u>INTERFACE_LINE_ATTRIBUTE6</u> <u>VARCHAR2(30)</u>
Sales Order lines and Return lines	Oracle Order Entry inserts SO_LINES.LINE_ID.
Freight charges	Oracle Order Entry inserts SO_FREIGHT_CHARGES.PICKING_HEADER_ID.
	<u>INTERFACE_LINE_ATTRIBUTE7</u> <u>VARCHAR2(30)</u>
Shipped order line	Oracle Order Entry inserts SO_PICKING_LINES.PICKING_LINE_ID
Return lines	Oracle Order Entry inserts 0 (zero).
Freight charges	Oracle Order Entry inserts 0 (zero).
	<u>INTERFACE_LINE_ATTRIBUTE8 - 15</u> <u>VARCHAR2(30)</u>
	Oracle Order Entry does not insert a value into this column.
	<u>BATCH_SOURCE_NAME</u> Not Null <u>VARCHAR2(50)</u>
	Oracle Order Entry enters the invoice source name you choose when you run the Receivables Interface program.

	<u>SET_OF_BOOKS_ID</u>	Not Null	<u>NUMBER (15)</u>
	Oracle Order Entry inserts the ID for the profile option, OE: Set of Books .		
	<u>LINE _TYPE</u>	Not Null	<u>VARCHAR2(20)</u>
Sales Order and Return lines	Oracle Order Entry inserts LINE .		
Freight charges	Oracle Order Entry inserts FREIGHT for shipment freight charges.		
	<u>DESCRIPTION</u>	Not Null	<u>VARCHAR2(240)</u>
Sales Order and Return lines	Oracle Order Entry inserts MTL_SYSTEM_ITEMS.DESCRPTION for the item.		
Freight charges	Oracle Order Entry inserts Freight Charge for shipment freight charges.		
	<u>CURRENCY_CODE</u>	Not Null	<u>VARCHAR2(30)</u>
	Oracle Order Entry inserts SO_HEADERS.CURRENCY_CODE.		
	<u>AMOUNT</u>		<u>NUMBER</u>
Sales Order and Return lines	Oracle Order Entry inserts a calculated amount (SO_LINES.SELLING_PRICE multiplied by SO_LINES.QUANTITY) based on the calculated quantity. Oracle Order Entry rounds the amount based on the minimum accounting unit and precision associated with the currency of the order.		
	The amount sign will match the sign on the quantity based on the value of RA_CUST_TRX_TYPES.CREATION_SIGN.		
Freight charges	Oracle Order Entry inserts SUM(SO_FREIGHT_CHARGES.AMOUNT) FROM SO_FREIGHT_CHARGES GROUP BY SO_FREIGHT_CHARGES. PICKING_HEADER_ID).		
	<u>CUST_TRX_TYPE_NAME</u>		<u>VARCHAR2(20)</u>
	Oracle Order Entry does not insert a value into this column.		

	<u>CUST_TRX_TYPE_ID</u>	<u>NUMBER(15)</u>
Sales Order lines	Oracle Order Entry inserts SO_ORDER_TYPES.CUST_TRX_TYPE_ID for the order type associated with the order. However, if there is a commitment on the order line, Oracle Order Entry does not insert a value into this column so that AutoInvoice can default the CUST_TRX_TYPE_ID from the commitment.	
Return lines	Oracle Order Entry inserts RA_CUST_TRX_TYPES.CREDIT_MEMO_TYPE_ID WHERE RA_CUST_TRX_TYPES.CUST_TRX_TYPE_ID = SO_ORDER_TYPES.CUST_TRX_TYPE_ID AND SO_ORDER_TYPES.ORDER_TYPE_ID = SO_HEADERS.ORDER_TYPE_ID AND SO_HEADERS.HEADER_ID = SO_LINES.HEADER_ID.	
Freight charges	Oracle Order Entry inserts SO_ORDER_TYPES.CUST_TRX_TYPE_ID associated with the order on the shipment.	
	<u>TERM_NAME</u>	<u>VARCHAR2(15)</u>
	Oracle Order Entry does not insert a value into this column.	
	<u>TERM_ID</u>	<u>NUMBER(15)</u>
Sales Order lines	Oracle Order Entry inserts SO_LINES.TERMS_ID.	
Return lines	Oracle Order Entry does not insert a value into this column.	
Freight charges	Oracle Order Entry inserts SO_HEADERS.TERMS_ID associated with the order on the shipment.	
	<u>ORIG_SYSTEM_BILL_</u> <u>CUSTOMER_REF</u>	<u>VARCHAR2(240)</u>
	Oracle Order Entry does not insert a value into this column.	
	<u>ORIG_SYSTEM_BILL_CUSTOMER_ID</u>	<u>NUMBER(15)</u>
	Oracle Order Entry inserts RA_ADDRESSES.CUSTOMER_ID FROM RA_ADDRESSES WHERE SO_HEADERS.INVOICE_TO_SITE_USE_ID	

= RA_SITE_USES.SITE_USE_ID
AND RA_SITE_USES.ADDRESS_ID
= RA_ADDRESSES.ADDRESS_ID.

ORIG_SYSTEM_BILL_
ADDRESS_REF **VARCHAR2(240)**

Oracle Order Entry does not insert a value into this column.

ORIG_SYSTEM_BILL_ADDRESS_ID **NUMBER(15)**

Oracle Order Entry inserts

RA_SITE_USES.ADDRESS_ID FROM RA_ADDRESSES
WHERE SO_HEADERS.INVOICE_TO_SITE_USE_ID
= RA_SITE_USES.SITE_USE_ID.

ORIG_SYSTEM_BILL_
CONTACT_REF **VARCHAR2(240)**

Oracle Order Entry does not insert a value into this column.

ORIG_SYSTEM_BILL_
CONTACT_ID **NUMBER(15)**

Oracle Order Entry inserts SO_HEADERS.INVOICE_TO_
CONTACT_ID.

ORIG_SYSTEM_SHIP_
CUSTOMER_REF **VARCHAR2(240)**

Oracle Order Entry does not insert a value into this column.

ORIG_SYSTEM_SHIP_CUSTOMER_ID **NUMBER(15)**

**Sales Order lines and
Return lines**

Oracle Order Entry inserts

RA_ADDRESSES.CUSTOMER_ID FROM RA_ADDRESSES
WHERE RA_ADDRESSES.ADDRESS_ID
= RA_SITE_USES.ADDRESS_ID
AND RA_SITE_USES.SITE_USE_ID
= NVL(SO_LINES.SHIP_TO_SITE_USE_ID,
SO_HEADERS.SHIP_TO_SITE_USE_ID).

Freight charges

Oracle Order Entry inserts


```

RA_ADDRESSES.CUSTOMER_ID FROM RA_ADDRESSES
WHERE RA_ADDRESSES.ADDRESS_ID
      = RA_SITE_USES.ADDRESS_ID
AND RA_SITE_USES.SITE_USE_ID
      = SO_HEADERS.SHIP_TO_SITE_USE_ID.

```

ORIG_SYSTEM_SHIP_
ADDRESS_REF **VARCHAR2(240)**

Oracle Order Entry does not insert a value into this column.

ORIG_SYSTEM_SHIP_ADDRESS_ID **NUMBER(15)**

**Sales Order lines and
Return lines**

Oracle Order Entry inserts

```

RA_SITE_USES.ADDRESS_ID FROM RA_ADDRESSES
WHERE RA_SITE_USES.SITE_USE_ID
      = NVL(SO_PICKING_LINES.SHIP_TO_SITE_USE_ID,
            NVL(SO_PICKING_HEADERS.SHIP_TO_SITE_USE_ID,
            NVL(SO_LINES.SHIP_TO_SITE_USE_ID,
            SO_HEADERS.SHIP_TO_SITE_USE_ID))).

```

Freight charges

Oracle Order Entry inserts

```

RA_SITE_USES.ADDRESS_ID FROM RA_ADDRESSES
WHERE RA_SITE_USES.SITE_USE_ID
      = NVL(SO_PICKING_HEADERS.SHIP_TO_SITE_USE_ID,
            SO_HEADERS.SHIP_TO_SITE_USE_ID).

```

ORIG_SYSTEM_SHIP_
CONTACT_REF **VARCHAR2(240)**

Oracle Order Entry does not insert a value into this column.

ORIG_SYSTEM_SHIP_CONTACT_ID **NUMBER(15)**

**Sales Order lines and
Return lines**

Oracle Order Entry inserts NVL(SO_LINES.SHIP_TO_CONTACT_ID,
SO_HEADERS.SHIP_TO_CONTACT_ID).

Freight charges

Oracle Order Entry inserts SO_HEADERS.SHIP_TO_CONTACT_ID.

ORIG_SYSTEM_SOLD_	
CUSTOMER_REF	VARCHAR2(240)

Oracle Order Entry does not insert a value into this column.

ORIG_SYSTEM_SOLD_	
CUSTOMER_ID	NUMBER(15)

Oracle Order Entry inserts SO_HEADERS.CUSTOMER_ID.

LINK_TO_LINE_ID	NUMBER(15)
------------------------	-------------------

Oracle Order Entry does not insert a value into this column.

LINK_TO_LINE_CONTEXT	VARCHAR2(30)
-----------------------------	---------------------

Oracle Order Entry does not insert a value into this column.

LINK_TO_LINE_	
ATTRIBUTE1 - 15	VARCHAR2(30)

Oracle Order Entry does not insert a value into these columns.

RECEIPT_METHOD_NAME	VARCHAR2(30)
----------------------------	---------------------

Oracle Order Entry does not insert a value into this column.

RECEIPT_METHOD_ID	NUMBER(15)
--------------------------	-------------------

Oracle Order Entry does not insert a value into this column.

CONVERSION_TYPE	Not Null	VARCHAR2(30)
------------------------	----------	---------------------

Oracle Order Entry inserts NVL(SO_HEADERS.CONVERSION_TYPE, 'User').

CONVERSION_DATE	DATE
------------------------	-------------

Oracle Order Entry inserts SO_HEADERS.CONVERSION_DATE.

CONVERSION_RATE	NUMBER
------------------------	---------------

Oracle Order Entry inserts SO_HEADERS.CONVERSION_RATE.

CUSTOMER_TRX_ID **NUMBER(15)**

Oracle Order Entry does not insert a value into this column.

TRX_DATE **DATE**

Oracle Order Entry does not insert a value into this column.

GL_DATE **DATE**

Oracle Order Entry does not insert a value into this column.

DOCUMENT_NUMBER **NUMBER(15)**

Oracle Order Entry does not insert a value into this column.

TRX_NUMBER **VARCHAR2(20)**

Oracle Order Entry does not insert a value into this column.
AutoInvoice determines a unique number for this transaction.

LINE_NUMBER **NUMBER(15)**

Oracle Order Entry does not insert a value into this column.

QUANTITY **NUMBER**

Oracle Order Entry inserts a calculated quantity based on the type of line being interfaced.

Sales Order lines The quantity will be either negative or positive depending on the value of RA_CUST_TRX_TYPES.CREATION_SIGN associated with the invoice type for the sales order. If RA_CUST_TRX_TYPES.CREATION_SIGN is N, then quantity passed is -1 multiplied by the quantity calculated.

Shipped order lines When picking lines exist for the order line:

STANDARD LINES

Quantity = SO_PICKING_LINES.SHIPPED_QUANTITY
- SO_PICKING_LINES.INVOICED_QUANTITY

PTO CONFIGURATION LINES - PTO MODEL

This is a multistep process:

1. Starting quantity = SO_LINES.SHIPPED_QUANTITY
– SO_LINES.INVOICED_QUANTITY
2. Next, look at all the required for revenue included items for the model and the required for revenue "children" lines of the model (option class, option items, models):

min child quantity =
 min (child SO_LINES.QUANTITY_TO_INVOICE/
 child SO_LINES.ORDERED_QUANTITY)
 * model SO_LINES.ORDERED_QUANTITY
 where the child line items are required
 for revenue in the bill of material for the model

min included item quantity =
 min (included item SO_PICKING_LINES.SHIPPED_QUANTITY/
 included item SO_PICKING_LINES.ORIGINAL_REQUESTED_
 QUANTITY)* model SO_LINES.ORDERED_QUANTITY
 where the included items are required for revenue in the
 bill of material for the model
3. Total quantity to invoice =
 minimum of (STARTING_QUANTITY,
 min child quantity,
 min included item quantity)
4. Quantity =
 minimum of (SO_PICKING_LINES.SHIPPED_QUANTITY
 for the model, total quantity to invoice)

– PTO CLASS

1. Starting quantity = SO_LINES.SHIPPED_QUANTITY
– SO_LINES.INVOICED_QUANTITY
2. Next, look at all the required for revenue included items for the class and the required for revenue "children" lines of the class (option class, option items)

min child quantity =
 min (child SO_LINES.QUANTITY_TO_INVOICE/
 child SO_LINES.ORDERED_QUANTITY)
 * class SO_LINES.ORDERED_QUANTITY
 where the child line items are required
 for revenue in the bill of material for the class

min included item quantity =
 min (included item SO_PICKING_LINES.SHIPPED_QUANTITY/
 included item SO_PICKING_LINES.ORIGINAL_REQUESTED_
 QUANTITY) * model SO_LINES.ORDERED_QUANTITY

where the included items are required for revenue in the BOM for the class

3. Total quantity to invoice = minimum of (STARTING_QUANTITY, min child quantity, min included item quantity)
4. Quantity =
minimum of (SO_PICKING_LINES.SHIPPED_QUANTITY for the class, total quantity to invoice)

- PTO OPTION ITEM OR KIT

1. Starting quantity = SO_LINES.SHIPPED_QUANTITY
- SO_LINES.INVOICED_QUANTITY
2. Next, look at all the required for revenue included items for the option item and the required for revenue included items of the option item "parent" line (option class)

min included item quantity =
min (included item SO_PICKING_LINES.SHIPPED_QUANTITY/
included item SO_PICKING_LINES.ORIGINAL_REQUESTED_QUANTITY) * option item SO_LINES.ORDERED_QUANTITY
where the included items are required for revenue in the bill of material for the option item or for the option item "parent" class

3. Total quantity to invoice = minimum of (STARTING_QUANTITY, min included item quantity)
4. Quantity = minimum of
(SO_PICKING_LINES.SHIPPED_QUANTITY for the class, total quantity to invoice)

ATO CONFIGURATION LINES

- ATO MODEL

Quantity =
SO_PICKING_LINES.SHIPPED_QUANTITY
- SO_PICKING_LINES.INVOICED_QUANTITY

- ATO CLASS

Quantity =
(class ORDERED_QUANTITY/model ordered quantity)
* model quantity

- ATO OPTION ITEM

Quantity =
(option item ordered quantity/model ordered quantity)
* model quantity

Non-Shipped order line Non-SHIP Cycle or Receivables Interface prior to Picking/Shipping:
Quantity = SO_LINES.ORDERED_QUANTITY
– SO_LINES.CANCELLED_QUANTITY

Return Lines Oracle Order Entry uses ORDERED_QUANTITY, SHIPPED_QUANTITY and INVOICED_QUANTITY to hold the values for the quantity authorized to be returned, the quantity received and accepted in Oracle Inventory and the quantity credited in Oracle Receivables, respectively. The value of SO_LINES.SHIPPED_QUANTITY is calculated in the RMA Inventory Interface program.

The quantity will be either negative or positive depending on the value of RA_CUST_TRX_TYPES.CREATION_SIGN associated with the credit memo type tied to the invoice type for return. If RA_CUST_TRX_TYPES.CREATION_SIGN is A or N, then the quantity passed is -1 multiplied by the quantity calculated.

Return line with receipt verification With RMA Inventory Receipt
Quantity = SO_LINES.SHIPPED_QUANTITY –
SO_LINES.INVOICED_QUANTITY

Return line without receipt verification Without RMA Inventory Receipt
Quantity = SO_LINES.ORDERED_QUANTITY –
SO_LINES.CANCELLED_QUANTITY

QUANTITY_ORDERED **NUMBER**

Sales Order lines and Return lines Oracle Order Entry inserts SO_LINES.ORDERED_QUANTITY.

Freight charges Oracle Order Entry does not insert a value into this column.

UNIT_SELLING_PRICE **NUMBER**

Sales Order lines and Return lines Oracle Order Entry inserts SO_LINES.SELLING_PRICE.

Freight charges Oracle Order Entry does not insert a value into this column.

	UNIT_STANDARD_PRICE	NUMBER
Sales Order lines and Return lines	Oracle Order Entry inserts SO_LINES.LIST_PRICE.	
Freight charges	Oracle Order Entry does not insert a value into this column.	
	PRINTING_OPTION	VARCHAR2(20)
	Oracle Order Entry does not insert a value into this column.	
	INTERFACE_STATUS	VARCHAR2(1)
	Oracle Order Entry does not insert a value into this column.	
	REQUEST_ID	NUMBER(15)
	Oracle Order Entry does not insert a value into this column.	
	RELATED_BATCH_ SOURCE_NAME	VARCHAR2(50)
	Oracle Order Entry does not insert a value into this column.	
	RELATED_TRX_NUMBER	VARCHAR2(20)
	Oracle Order Entry does not insert a value into this column.	
	RELATED_CUSTOMER_ TRX_ID	NUMBER(15)
	Oracle Order Entry does not insert a value into this column.	
	PREVIOUS_CUSTOMER_ TRX_ID	NUMBER(15)
	Oracle Order Entry does not insert a value into this column.	
	CREDIT_METHOD_ FOR_ACCT_RULE	VARCHAR2(30)
Sales Order lines	Oracle Order Entry does not insert a value into this column.	

Return lines Oracle Order Entry inserts the accounting credit method, NVL(SO_ORDER_TYPES.ACCOUNTING_CREDIT_METHOD_CODE, 'LIFO') associated with the order type.

CREDIT_METHOD_FOR_INSTALLMENTS **VARCHAR2(30)**

Sales Order lines Oracle Order Entry does not insert a value into this column.

Return lines Oracle Order Entry inserts the invoicing credit method, NVL(SO_ORDER_TYPES.INVOICING_CREDIT_METHOD_CODE, 'LIFO') associated with the order type.

REASON_CODE **VARCHAR2(30)**

Sales Order lines Oracle Order Entry does not insert a value into this column.

Return lines Oracle Order Entry inserts SO_LINES.TRANSACTION_REASON_CODE.

TAX_RATE **NUMBER**

Oracle Order Entry does not insert a value into this column.

TAX_CODE **VARCHAR2(50)**

Oracle Order Entry does not insert a value into this column.

TAX_PRECEDENCE **NUMBER(15)**

Oracle Order Entry does not insert a value into this column.

EXCEPTION_ID **NUMBER(15)**

Oracle Order Entry does not insert a value into this column.

EXEMPTION_ID **NUMBER(15)**

Oracle Order Entry does not insert a value into this column.

SHIP_DATE_ACTUAL **DATE**

Non-Shipped order line Oracle Order Entry does not insert a value into this column.

Shipped order line	Oracle Order Entry inserts SO_PICKING_HEADERS.DATE_SHIPPED.
Freight charges	Oracle Order Entry inserts SO_PICKING_HEADERS.DATE_SHIPPED.
	FOB_POINT VARCHAR2(20)
	Oracle Order Entry only populates this column if the order line being invoiced has been shipped.
Non-Shipped order line	Oracle Order Entry does not insert a value into this column.
Shipped order line	Oracle Order Entry inserts SO_HEADERS.FOB_CODE.
Freight charges	Oracle Order Entry inserts SO_HEADERS.FOB_CODE FROM SO_HEADERS, SO_PICKING_HEADERS, SO_FREIGHT_CHARGES WHERE SO_HEADERS.HEADER_ID = SO_PICKING_HEADERS.ORDER_HEADER_ID AND SO_PICKING_HEADERS.PICKING_HEADER_ID = SO_FREIGHT_CHARGES.PICKING_HEADER_ID.
	SHIP_VIA VARCHAR2(20)
Non-Shipped order line	Oracle Order Entry does not insert a value into this column.
Shipped order line	Oracle Order Entry inserts SO_PICKING_HEADERS.SHIP_METHOD_CODE.
Freight charges	Oracle Order Entry inserts SO_PICKING_HEADERS.SHIP_METHOD_CODE.
	WAYBILL_NUMBER VARCHAR2(50)
Non-Shipped order line	Oracle Order Entry does not insert a value into this column.
Shipped order line	Oracle Order Entry inserts SO_PICKING_HEADERS.WAYBILL_NUM.
Freight charges	Oracle Order Entry inserts SO_PICKING_HEADERS.WAYBILL_NUM.

INVOICING_RULE_NAME **VARCHAR2(30)**

Oracle Order Entry does not insert a value into this column.

INVOICING_RULE_ID **NUMBER(15)**

Sales Order lines Oracle Order Entry inserts DECODE(ACCOUNTING_RULE_ID,1,
NULL,INVOICING_RULE_ID).

Note: For more information on how Oracle Order Entry determines the accounting rule used in the decode, refer to the section at the beginning of this essay, **Accounting and Invoicing Rules** (See page 6 – 18).

Return lines Oracle Order Entry does not insert a value into this column.

Freight charges Oracle Order Entry inserts DECODE(ACCOUNTING_RULE_ID,1,
NULL,SO_HEADERS.INVOICING_RULE_ID).

ACCOUNTING_RULE_NAME **VARCHAR2(30)**

Oracle Order Entry does not insert a value into this column.

ACCOUNTING_RULE_ID **NUMBER(15)**

Sales Order lines Oracle Order Entry inserts DECODE(ACCOUNTING_RULE_ID,1,
NULL,ACCOUNTING_RULE_ID).

Note: For more information on how Oracle Order Entry determines the accounting rule used in the decode, refer to the section at the beginning of this essay, **Accounting and Invoicing Rules** (See page 6 – 18).

Return lines Oracle Order Entry does not insert a value into this column.

Freight charges Oracle Order Entry inserts DECODE(ACCOUNTING_RULE_ID,1,
NULL,SO_HEADERS.ACCOUNTING_RULE_ID).

ACCOUNTING_RULE_
DURATION **NUMBER(15)**

Oracle Order Entry does not insert a value into this column.

RULE_START_DATE **DATE**

Oracle Order Entry does not insert a value into this column.

Sales Order lines and Return lines Freight charges	PRIMARY_SALESREP_	NUMBER	VARCHAR2(30)
	Oracle Order Entry does not insert a value into this column.		
	PRIMARY_SALESREP_ID	NUMBER	(15)
	Oracle Order Entry inserts SO_HEADERS.SALESREP_ID.		
	SALES_ORDER	VARCHAR2	(50)
	Oracle Order Entry inserts SO_HEADERS.ORDER_NUMBER.		
	SALES_ORDER_LINE	VARCHAR2	(30)
	Oracle Order Entry inserts SO_LINES.LINE_NUMBER.		
	Oracle Order Entry does not insert a value into this column.		
	SALES_ORDER_DATE	DATE	
	Oracle Order Entry inserts SO_HEADERS.DATE_ORDERED.		
	SALES_ORDER_SOURCE	VARCHAR2	(50)
	Oracle Order Entry inserts the value you have for the profile option, OE: Source Code.		
	SALES_ORDER_REVISION	NUMBER	
	Oracle Order Entry does not insert a value into this column.		
	PURCHASE_ORDER	VARCHAR2	(50)
	Oracle Order Entry inserts SO_HEADERS.PURCHASE_ ORDER_NUM.		
	PURCHASE_ORDER_REVISION	VARCHAR2	(50)
	Oracle Order Entry does not insert a value into this column.		
	PURCHASE_ORDER_DATE	DATE	
	Oracle Order Entry does not insert a value into this column.		

AGREEMENT_NAME **VARCHAR2(30)**

Oracle Order Entry does not insert a value into this column.

AGREEMENT_ID **NUMBER(15)**

Sales Order lines Oracle Order Entry inserts SO_HEADERS.AGREEMENT_ID.

Return lines For return lines with a purchase order or sales order reference, Oracle Order Entry inserts the AGREEMENT_ID from the referenced order header. For return lines with an invoice reference, Oracle Order Entry does not insert a value into this column.

Freight charges Oracle Order Entry does not insert a value into this column.

MEMO_LINE_NAME **VARCHAR2(50)**

Oracle Order Entry does not insert a value into this column.

MEMO_LINE_ID **NUMBER(15)**

Oracle Order Entry does not insert a value into this column.

INVENTORY_ITEM_ID **NUMBER(15)**

Sales Order lines and Oracle Order Entry inserts SO_LINES.INVENTORY_ITEM_ID.

Return lines

Freight charges Oracle Order Entry does not insert a value into this column.

MTL_SYSTEM_ITEMS_
SEG1 – 20 **VARCHAR2(30)**

Oracle Order Entry does not insert a value into this column.

REFERENCE_LINE_ID **NUMBER(15)**

Sales Order lines Oracle Order Entry inserts SO_LINES.COMMITMENT_ID.

Return lines Oracle Order Entry inserts SO_LINES.CREDIT_TO_LINE_ID.

Freight charges Oracle Order Entry does not insert a value into this column.

REFERENCE_LINE_CONTEXT **VARCHAR2(30)**

Oracle Order Entry does not insert a value into this column.

	<u>REFERENCE_LINE_</u>	
	<u>ATTRIBUTE(1-15)</u>	<u>VARCHAR2(30)</u>
	Oracle Order Entry does not insert a value into this column.	
	<u>TERRITORY_ID</u>	<u>NUMBER(15)</u>
	Oracle Order Entry does not insert a value into this column.	
	<u>TERRITORY_SEGMENT1 – 20</u>	<u>VARCHAR2(25)</u>
	Oracle Order Entry does not insert a value into this column.	
	<u>ATTRIBUTE_CATEGORY</u>	<u>VARCHAR2(30)</u>
Sales Order lines and Return lines	Oracle Order Entry inserts SO_LINES.CONTEXT.	
Freight charges	Oracle Order Entry inserts SO_FREIGHT_CHARGES.CONTEXT.	
	<u>ATTRIBUTE1 – 15</u>	<u>VARCHAR2(150)</u>
Sales Order lines and Return lines	Oracle Order Entry inserts SO_LINES.ATTRIBUTE1-15.	
Freight charges	Oracle Order Entry inserts SO_FREIGHT_CHARGES.ATTRIBUTE1-15.	
	<u>HEADER_ATTRIBUTE_CATEGORY</u>	<u>VARCHAR2(30)</u>
Sales Order lines and Return lines	Oracle Order Entry inserts SO_HEADERS.CONTEXT.	
Freight charges	Oracle Order Entry does not insert a value into this column.	
	<u>HEADER_ATTRIBUTE(1-15)</u>	<u>VARCHAR2(150)</u>
Sales Order lines and Return lines	Oracle Order Entry inserts SO_HEADERS.ATTRIBUTE1 – 15.	
Freight charges	Oracle Order Entry does not insert a value into this column.	
	<u>COMMENTS</u>	<u>VARCHAR2(240)</u>
	Oracle Order Entry does not insert a value into this column.	

INTERNAL_NOTES **VARCHAR2(240)**

Oracle Order Entry does not insert a value into this column.

**INITIAL_CUSTOMER_
TRX_ID** **NUMBER(15)**

Oracle Order Entry does not insert a value into this column.

**USSGL_TRANSACTION_
CODE_CONTEXT** **VARCHAR2(30)**

Oracle Order Entry does not insert a value into this column.

**USSGL_TRANSACTION_
CODE** **VARCHAR2(30)**

Oracle Order Entry does not insert a value into this column.

ACCTD_AMOUNT **NUMBER**

Oracle Order Entry does not insert a value into this column.

**CUSTOMER_BANK_
ACCOUNT_ID** **NUMBER(15)**

Oracle Order Entry does not insert a value into this column.

**CUSTOMER_BANK_
ACCOUNT_NAME** **VARCHAR2(25)**

Oracle Order Entry does not insert a value into this column.

UOM_CODE **VARCHAR2(3)**

**Sales Order lines and
Return lines**

Oracle Order Entry inserts SO_LINES.UNIT_CODE.

Freight charges

Oracle Order Entry does not insert a value into this column.

UOM_NAME **VARCHAR2(25)**

Oracle Order Entry does not insert a value into this column.

DOCUMENT_NUMBER_	
SEQUENCE_ID	NUMBER(15)

Oracle Order Entry does not insert a value into this column.

REASON_CODE_NAME	VARCHAR2(30)
-------------------------	---------------------

Oracle Order Entry does not insert a value into this column.

VAT_TAX_ID	NUMBER(15)
-------------------	-------------------

Oracle Order Entry does not insert a value into this column.

LOCATION_RATE_ID	NUMBER(15)
-------------------------	-------------------

Oracle Order Entry does not insert a value into this column.

REASON_CODE_MEANING	VARCHAR2(80)
----------------------------	---------------------

Oracle Order Entry does not insert a value into this column.

LAST_PERIOD_TO_CREDIT	NUMBER
------------------------------	---------------

Oracle Order Entry does not insert a value into this column.

PAYING_CUSTOMER_ID	NUMBER(15)
---------------------------	-------------------

Oracle Order Entry does not insert a value into this column.

PAYING_SITE_USE_ID	NUMBER(15)
---------------------------	-------------------

Oracle Order Entry does not insert a value into this column.

TAX_EXEMPT_FLAG	VARCHAR2(1)
------------------------	--------------------

Oracle Order Entry inserts SO_HEADERS.TAX_EXEMPT_FLAG for order lines.

SALES_TAX_ID	NUMBER(15)
---------------------	-------------------

Oracle Order Entry does not insert a value into this column.

CREATED_BY **NUMBER(15)**

Oracle Order Entry enters an identification number to identify the user who created the record.

Validation: None

CREATION_DATE **DATE**

Oracle Order Entry enters the creation date.

Validation: Standard Date Validation

LAST_UPDATED_BY **NUMBER(15)**

Oracle Order Entry enters an identification number to identify the user who created or who most recently modified the record.

Validation: None

LAST_UPDATE_DATE **DATE**

Oracle Order Entry enters the current date when a record is updated.

Validation: Standard Date Validation

LOCATION_SEGMENT_ID **NUMBER(15)**

Oracle Order Entry does not insert a value into this column.

TAX_EXEMPT_REASON_CODE **VARCHAR2(30)**

Oracle Order Entry inserts SO_HEADERS.TAX_EXEMPT_REASON_CODE for order lines.

TAX_EXEMPT_NUMBER **VARCHAR2(80)**

Oracle Order Entry inserts SO_HEADERS.TAX_EXEMPT_NUM for order lines.

TAX_EXEMPT_REASON_CODE_MEANING **VARCHAR2(80)**

Oracle Order Entry does not insert a value into this column.

RA_INTERFACE_SALESCREDITS

Oracle Order Entry inserts one row for each sales credit row according to the following hierarchy:

1. insert sales credits associated with the line; if none exists but the line is part of a configuration (ITEM_TYPE_CODE is **CLASS**, **KIT** or **STANDARD** and OPTION_FLAG is **Y**) then...
2. insert sales credits associated with the model "parent" line; if none exists then...
3. insert sales credits associated with the order header.

INTERFACE_SALESCREDIT_ID	NUMBER(15)
---------------------------------	-------------------

Oracle Order Entry does not insert a value into this column.

INTERFACE_LINE_ID	NUMBER(15)
--------------------------	-------------------

Oracle Order Entry does not insert a value into this column.

INTERFACE_LINE_CONTEXT	VARCHAR2(30)
-------------------------------	---------------------

Oracle Order Entry inserts your setting for the profile option,
OE: Source Code.

INTERFACE_LINE_ATTRIBUTE1	VARCHAR2(30)
----------------------------------	---------------------

Oracle Order Entry inserts SO_HEADERS. ORDER_NUMBER.

INTERFACE_LINE_ATTRIBUTE2	VARCHAR2(30)
----------------------------------	---------------------

Oracle Order Entry inserts SO_ORDER_TYPES.NAME.

INTERFACE_LINE_ATTRIBUTE3	VARCHAR2(30)
----------------------------------	---------------------

Shipped order line	Oracle Order Entry inserts SUBSTR(SO_PICKING_HEADERS.PICK_SLIP_NUMBER, 1, 30); this is the first thirty characters of a fifty character field.
---------------------------	--

Non-Shipped order line	Oracle Order Entry inserts 0 (zero).
-------------------------------	--------------------------------------

Return lines	Oracle Order Entry inserts 0 (zero).
---------------------	--------------------------------------

	<u>INTERFACE_LINE_ATTRIBUTE4</u>	<u>VARCHAR2(30)</u>
Shipped order line	Oracle Order Entry inserts SUBSTR(SO_PICKING_HEADERS.WAYBILL_NUMBER, 1, 30).	
Non-Shipped order line	Oracle Order Entry inserts 0 (zero).	
Return lines	Oracle Order Entry inserts 0 (zero).	
	<u>INTERFACE_LINE_ATTRIBUTE5</u>	<u>VARCHAR2(30)</u>
	Oracle Order Entry inserts the number of times the order, freight charge, or return line has been interfaced for invoice or credit.	
	<u>INTERFACE_LINE_ATTRIBUTE6</u>	<u>VARCHAR2(30)</u>
Sales Order lines and Return lines	Oracle Order Entry inserts SO_LINES.LINE_ID.	
	<u>INTERFACE_LINE_ATTRIBUTE7</u>	<u>VARCHAR2(30)</u>
Shipped order line	Oracle Order Entry inserts SO_PICKING_LINES.PICKING_LINE_ID	
Return lines	Oracle Order Entry inserts 0 (zero).	
	<u>INTERFACE_LINE_ATTRIBUTE8 – 15</u>	<u>VARCHAR2(30)</u>
	Oracle Order Entry does not insert a value into this column.	
	<u>SALESREP_NUMBER</u>	<u>VARCHAR2(30)</u>
	Oracle Order Entry does not insert a value into this column.	
	<u>SALESREP_ID</u>	<u>NUMBER(15)</u>
	Oracle Order Entry inserts SO_SALES_CREDIT.SALESREP_ID.	
	<u>SALES_CREDIT_TYPE_NAME</u>	<u>VARCHAR2(30)</u>
	Oracle Order Entry does not insert a value into this column.	

<u>SALES_CREDIT_TYPE_ID</u>	<u>NUMBER(15)</u>
------------------------------------	--------------------------

Oracle Order Entry inserts SO_SALES_CREDIT.SALES_CREDIT_TYPE_ID.

<u>SALES_CREDIT_</u>	
<u>AMOUNT_SPLIT</u>	<u>NUMBER</u>

Oracle Order Entry does not insert a value into this column.

<u>SALES_CREDIT_</u>	
<u>PERCENT_SPLIT</u>	<u>NUMBER</u>

Oracle Order Entry inserts SO_SALES_CREDIT.PERCENT.

<u>INTERFACE_STATUS</u>	<u>VARCHAR2(1)</u>
--------------------------------	---------------------------

Oracle Order Entry does not insert a value into this column.

<u>REQUEST_ID</u>	<u>NUMBER(15)</u>
--------------------------	--------------------------

Oracle Order Entry does not insert a value into this column.

<u>ATTRIBUTE_CATEGORY</u>	<u>VARCHAR2(30)</u>
----------------------------------	----------------------------

Oracle Order Entry inserts SO_SALES_CREDIT.CONTEXT.

<u>ATTRIBUTE1 – 15</u>	<u>VARCHAR2(150)</u>
-------------------------------	-----------------------------

Oracle Order Entry inserts SO_SALES_CREDIT.ATTRIBUTE1 – 15.

Integrating Oracle Order Entry Using OrderImport

OrderImport is a powerful, flexible open interface that lets you capture and validate order data from external systems.

Basic Needs

OrderImport provides the features you need to satisfy the following basic integration needs:

- Import data from a variety of environments, including your own order entry systems.
- Import complete (historical) orders.
- Import incomplete orders and finish them using the Enter Orders form.
- Import booked orders.
- Import changes to existing orders.
- Place demand or reserve inventory for orders automatically.
- Optionally price order lines using the same pricing rules as online entry, supporting individual customer price lists and discounts.
- Import quota and nonquota sales credit information for orders and order lines.
- Enforce holds and perform credit checking on any imported orders or changes.
- Import line schedule detail information with each order and order line.
- Enforce security rules and utilize defaulting with imported orders.
- Import internal requisition orders from Oracle Purchasing.
- Apply automatic notes.
- Review your imported data in a report. Not only can you identify which data has been successfully imported, you can identify any errors which may have occurred during the import process.

Order Importing

OrderImport lets you capture order information using a specialized system and import them into Oracle Order Entry for processing. You can import orders with any entry status, including **Booked**. Imported orders can be queried and modified using the Enter Orders form in Oracle Order Entry. If an order is imported with an entry status of Booked, it will automatically be eligible to progress to the next step of its order cycle when the import is complete.

Historical Data Conversion (Complete orders)

OrderImport also allows you to import completed orders from your previous order entry system, making the transition from your old application to Oracle Order Entry as smooth as possible. Complete orders are considered for historical purposes only and can have any entered state and the status for the **Complete Order** cycle action is **Closed**. Specify complete orders by entering **Y** in SO_HEADERS_INTERFACE.COMPLETE_FLAG.

Entered State

You can import both current and complete orders with entered states of **Entered**, **Partial**, or **Booked**. OrderImport ensures that all required fields for entry or booking are validated appropriately as the orders are imported. If any of the required fields for a booked order being imported are not supplied, Oracle Order Entry rejects the order. Refer to the Reporting (see page 6 – 54) section for more information on how to report on errors.

Order Cycles

You can import an order within any valid order cycle. The order must be at the initial action of Enter or the final action of Complete. Orders transferred through batch order entry cannot be in the middle of a cycle.

Configurations

You can import all orderable items of a configuration, such as models, classes and options. OrderImport, using the information you supply in the interface tables, will validate the configuration against the bill of material and create related order lines for the model, classes and

options. If you are importing assemble-to-order (ATO) configurations, you cannot import the mandatory standard components required to build the configuration. Oracle Work in Process will select these automatically, from the bill of material, when the work order is opened. If you are importing pick-to-order (PTO) configurations, you cannot import the included items. Oracle Order Entry will select these automatically, from the bill of material, when the included items are exploded. The point in the cycle when included items are exploded is defined by the profile option **OE: Included Item Freeze Method**.

Changes to Imported Orders

You can import changes to orders that have already been imported by passing all changed and unchanged information through OrderImport. You can insert, update and delete orders, order lines, price adjustments and sales credits.

Order Scheduling

OrderImport lets you demand or reserve orders as they are imported, using the same rules as online order entry. If the scheduling request is unsuccessful on an imported order with an entry status of **Booked**, the order will be imported, and the scheduling exception is reported in the OrderImport Processing Results Report. You can then update the order on the Enter Order or Schedule Orders form.



Scheduling Orders
(*Oracle Order Entry Reference Manual*)



OrderImport Processing Results Report
(*Oracle Order Entry Reference Manual*)

Schedule Details

OrderImport allows you to import scheduling details for an order line by inserting data into the SO_LINE_DETAILS_INTERFACE table. For example, you may want to specify a subinventory or lot when you reserve an order line, or you can specify multiple detail records to split an order line between different warehouses or schedule dates.

Pricing

You can indicate whether you want to manually enter prices for imported orders or have Oracle Order Entry automatically price order lines using the CALCULATE_PRICE column in the SO_LINES_

INTERFACE table. We recommend that you either use completely automatic pricing or completely manual pricing with your imported orders. If you want to use the automatic method, you should set SO_LINES_INTERFACE.CALCULATE_PRICE to Yes, and define all your discounts as line level and automatic. If you want to use the manual method, you should set SO_LINES_INTERFACE.CALCULATE_PRICE to No and import order lines with the list price, selling price, and any price adjustments. In this case, you should define all your discounts as line level, overridable, and not automatic.

If you are using the manual method, Oracle Order Entry lets you adjust the selling price of items through OrderImport by inserting data in the SO_PRICE_ADJUSTMENTS_INTERFACE table. If you are importing an order line with a list and selling price that are not the same, you *must* insert a valid price adjustment. The discount amount or percent of the referenced discount must either equal the difference between list and selling or have the Allow Override field set to Yes. You can enter multiple price adjustments for an order line as long as each price adjustment uses a unique discount and the total of the price adjustments equals the difference between the list and selling price.



Attention: Line level discounts that are defined as automatic are automatically applicable to imported orders that have Yes in the CALCULATE_PRICE column, so if you use automatic discounts, you do not need to import them with such order lines.

CALCULATE_PRICE - YES		CALCULATE_PRICE - NO	
List Price	Ignored	List Price	Required
Selling Price	Ignored	Selling Price	Required
Discount	Optional	Discount	Required if List Price \neq Selling Price
Note: Imported discounts must be overridable if you want to import a percent value that is different from that defined on the discount.			

Table 6 – 7 (Page 1 of 1)

Taxation and Tax Exemptions

OrderImport allows you to indicate the tax status of orders as they are imported, using the same rules as online order entry through the TAX_EXEMPT_FLAG, TAX_EXEMPT_NUM, and

TAX_EXEMPT_REASON_CODE columns in the SO_HEADERS_INTERFACE table. The TAX_EXEMPT_FLAG allows you to indicate one of the following:

- this order is exempt for a normally taxable customer site and/or item
- taxable for a normally non-taxable customer and/or item
- taxation should be based on existing exemption rules

If you choose to Exempt the order from taxation, you can enter a valid certificate number for the ship to customer or enter a new, unapproved exemption certificate number in the TAX_EXEMPT_NUM column. If you choose to Exempt the order from taxation, you must enter a TAX_EXEMPT_REASON_CODE. You can override the default tax code for an order line by specifying a value for TAX_CODE in the SO_LINES_INTERFACE table. If OrderImport determines that the order line is not taxable, the TAX_CODE column value in SO_LINES_INTERFACE will be preserved, but the imported order line will not have a tax code.

Sales Credits

Oracle Order Entry lets you enter sales credit information through OrderImport by inserting data in the SO_SALES_CREDITS_INTERFACE table. If the salesperson in SO_HEADERS_INTERFACE.SALESREP_ID receives 100% sales credit for an open order, you do not need to insert sales credits. However, if you want to apply sales credits to historical (closed) orders, split sales credits or assign different credits to each order line, you can use the SO_SALES_CREDITS_INTERFACE table to communicate this information.

The total of the revenue credit must equal 100 percent. Nonrevenue credit can be any percentage of the order line or order.

Agreements

You can include an agreement name if you want to order against a customer agreement for an entire order. All order lines of the order would use this agreement.



Define Agreements
(*Oracle Order Entry Reference Manual*)

Order Holds and Credit Checking

Oracle Order Entry automatically applies any appropriate, user defined customer or items holds to an imported order. This lets you hold imported orders for review, just as you do orders entered through the Enter Order form. Oracle Order Entry also performs credit checking on all imported orders, according to the credit checking rules you define.



Automatic Credit Checking of Orders
Managing Process Exceptions
(*Oracle Order Entry Reference Manual*)

Security Rules and Standard Value Rule Sets

OrderImport checks the security rules you have defined for your Oracle Order Entry application to assure that any updates you make to imported orders are acceptable by your security standards.

Oracle Order Entry allows you use standard value rule sets with OrderImport for defaulting information to the fields on the Enter Orders form in the same way as for orders entered directly. Standard value rule sets are assigned to the order type.



Controlling Changes to Orders and Returns
Using Standard Value Rule Sets
(*Oracle Order Entry Reference Manual*)

Internal Sales Orders

Oracle Purchasing uses OrderImport to transfer requisitions for internally sourced products to Oracle Order Entry. Once imported, these internal sales orders are processed like regular sales orders.



Internal Requisitions
(*Oracle Order Entry Reference Manual*)

Notes

Oracle Order Entry applies any automatic standard notes to imported orders that meet your automatic note criteria. Once an order is imported, you can edit these notes from the Enter Orders form as you would on a regular order.

Reporting

Each time you run OrderImport, Oracle Order Entry automatically generates the OrderImport Processing Results Report, which tells you the total number of orders imported successfully and any warnings about scheduling. Scheduling warnings include insufficient quantity on hand to reserve an item. If an order line has validation errors, OrderImport rejects the entire order. You can view this report from the View Reports or View Requests form (View Report option) and indicate either the concurrent request ID or the program name.

Oracle Order Entry reports all errors that occur during the import process in the OrderImport Process Exception Report. You can then correct any errors and successfully reimport your data.



View Requests

View Reports

(Oracle Applications User's Guide)



OrderImport Process Exception Report

OrderImport Processing Results Report

(Oracle Order Entry Reference Manual)

Understanding the Interface Tables

Oracle Order Entry uses five *Oracle* tables in which OrderImport receives data you import from other systems.

- SO_HEADERS_INTERFACE
- SO_LINES_INTERFACE
- SO_LINE_DETAILS_INTERFACE
- SO_PRICE_ADJUSTMENTS_INTERFACE
- SO_SALES_CREDITS_INTERFACE

When OrderImport receives data, it validates and converts your import data into orders within Oracle Order Entry. These interface tables are organized by columns in which Oracle Order Entry categorizes and stores specific data. For example, your original order number information is stored in the column called ORIGINAL_SYSTEM_REFERENCE in the table SO_HEADERS. (SO_HEADERS.ORIGINAL_SYSTEM_REFERENCE)

Prerequisites

Order Import expects that information such as customers, agreements, items, and salespersons will be defined before you interface records referencing them. To ensure that OrderImport works for you, you should prepare Oracle Order Entry for any new data that you want OrderImport to import. Therefore, before importing orders, you should:

Define QuickCodes

Add QuickCodes to Oracle Order Entry if your feeder systems use QuickCode values not yet defined in Oracle Order Entry.

Define Sales Credit Types

Add sales credit types to Oracle Order Entry if your feeder systems use sales credit types not yet defined in Oracle Order Entry.

Define Items

Add items to Inventory using the Define Items form if your feeder systems use items not yet defined.

Define Configurations

Define configurations in Oracle Order Entry if your feeder systems use configurations not yet defined in Oracle Order Entry. You first need to define each model, option class, option item, standard component, and included item as an item. Then use the Define Bill of Material form to set up your configuration structures.

Define Price Lists

Add price lists to Oracle Order Entry if your feeder systems use price lists not yet defined in Oracle Order Entry. If you import the list and selling price for your order lines you do not need to add each item to a price list, you can just reference a price list name that has no lines. However, you cannot add price adjustments to an order line without the item being on the price list.

Define Discounts

Add discounts to Oracle Order Entry if your feeder systems use discounts not yet defined in Oracle Order Entry. If you specify that

you want to manually calculate prices for imported orders (SO_LINES_INTERFACE.CALCULATE_PRICE is No), and you insert a list price that is different from the selling price, you also need to reference an existing discount to capture the difference. You can specify line level discounts with the Automatic field set to No. The total difference between the list and selling price must be represented by price adjustments. Also, if you want to apply order level discounts to imported order, you need to import the discount through the SO_PRICE_ADJUSTMENTS_INTERFACE table, because order level price adjustments are not automatically applicable.



Overview of Pricing
(*Oracle Order Entry Reference Manual*)

Define Order Types

Add order types to Oracle Order Entry if your feeder systems use order types not yet defined in Oracle Order Entry. If your order type references a standard value rule set, defaulting will automatically occur as your order is imported.

Define OrderImport Sources

Define OrderImport sources and determine whether to use IDs. (Refer to the Using IDs section on page 6 – 66.

Enter Customers and Customer Sites

Add customers and customer addresses to Oracle Order Entry if your feeder systems use customers not yet defined in Oracle Order Entry. If you are using a feeder system to add new customers, run the Customer Interface program before running OrderImport.

Define Salespersons

Add salespersons to Oracle Order Entry if your feeder systems use salespersons not yet defined in Oracle Order Entry.

Define Customer Agreements

Add customer agreements to Oracle Order Entry if your feeder systems use customer agreements not yet defined in Oracle Order Entry.

Standard Value Rule Sets

Define standard value rule sets to default information to the fields on the Enter Orders form based on other information you enter. Assign your standard value rule sets to order types.



Overview of Setup
(*Oracle Order Entry Reference Manual*)

Setting Up Oracle Order Entry
(*Oracle Financials and Oracle Government Financials Implementation Manual*)



Integrating Customer Information using Oracle Receivables Applications Customer Interface (See page 10 – 143)



Define Order Entry QuickCodes
Define Receivables QuickCodes
Define QuickCodes
Define Sales Credit Types
Define Price Lists
Define Discounts
Define Order Type
Define OrderImport Sources
Define Salespersons
Enter Customer Information
Define Agreements
Define Standard Value Rule Set
(*Oracle Order Entry Reference Manual*)

Define Item
(*Oracle Inventory Reference Manual*)

Define Bill Of Material
(*Oracle Bills of Material Reference Manual*)

Importing Data From Your Feeder System

OrderImport receives data from an import program which your onsite MIS personnel can develop for you; or you may wish to use an Oracle consultant. Your import program must convert data from your feeder system into a standard data format that OrderImport can read. OrderImport can then convert your import data into Oracle Order Entry orders.

The type of environment from which you import data determines the type of import program you need, as well as the import utility you should use to write your program. SQL*Loader is a powerful and easy to use tool you can use to write such a program. However, depending on the complexity of your needs, there are a number of programming languages available to you. You can also use embedded 3GL Oracle tools like Pro*C, Pro*COBOL and Pro*Fortran to write your program.

Identifying Source IDs

You can find the ORDER_SOURCE_ID in the SO_ORDER_SOURCES table for each OrderImport source that you define. Use the ORDER_SOURCE_ID when importing orders.

Assigning Values for System Items Flexfield Segments

Depending on the ORDER_SOURCE_ID you choose, the SO_ORDER_SOURCES.USE_IDS flag is either Y or N. If the USE_IDS flag is Y, the id columns are used. If the USE_IDS flag is N, NAME or SEGMENT columns are used.

If your USE_IDS flag is set to N, assign a System Items Flexfield (Item Flexfield) value to as many segments as you have enabled. As long as you enter the segments in the correct sequence, OrderImport knows which segment columns to place the values in. For example, if you enabled six Item Flexfield segments in Oracle Inventory, then you can assign values to columns SEGMENT1 through SEGMENT6. If you enabled only four Item Flexfield segments, then you can assign values to columns SEGMENT1 through SEGMENT4.

MTL_SYSTEM_ITEMS column	Flexfield Sequence	SO_LINES_INTERFACE column
SEGMENT1	1	INVENTORY_ITEM_SEGMENT1
SEGMENT20	2	INVENTORY_ITEM_SEGMENT2
SEGMENT13	3	INVENTORY_ITEM_SEGMENT3
SEGMENT5	4	INVENTORY_ITEM_SEGMENT4
SEGMENT17	5	INVENTORY_ITEM_SEGMENT5
SEGMENT3	6	INVENTORY_ITEM_SEGMENT6

Table 6 – 8 (Page 1 of 1)

Load valid enabled segment values for your enabled segments into the OrderImport tables. The segment values must already be defined. Made sure you specify the segment value correctly. For example, the value **01** is not the same as the value **1**. OrderImport does not allow null values in enabled segments.



Key Flexfields in Oracle Applications (System Items)
(*Oracle Applications Flexfields Manual*)

Assigning Values to Not Null Columns

You must assign values to all Not Null columns in the OrderImport tables in order for OrderImport to successfully convert your import data into orders.

Assigning Values to Optional Columns

You can enter optional column values for several columns in the OrderImport tables. Entering values in these optional columns gives you maximum control over the way OrderImport handles your data.

Importing Configurations

Oracle Order Entry's OrderImport lets you easily import models or configurations, option classes, and option items. Standard components and included items are not imported, but Oracle Order Entry derives them from the associated bill of material (BOM). Only Oracle Order Entry can explode a bill of material. The bill of material structure for an order line (SO_LINES_INTERFACE.ORIGINAL_SYSTEM_LINE_REFERENCE) is determined by the combination of PARENT_LINE_REF, LINK_TO_LINE_REF, ITEM_TYPE_CODE, and OPTION_FLAG columns in the SO_LINES_INTERFACE table.

The following table shows an example of a configuration that is four levels deep with option classes and models imbedded at different levels, and how to use the columns listed above to explain the bill structure to Oracle Order Entry. Refer to the Bills of Material and Configure to Order topical essays in the *Oracle Order Entry Reference Manual* for more information and further examples.

Level		ORIGINAL_SYSTEM_ LINE_REFERENCE	PARENT_LINE_REF	LINK_TO_LINE_REF	ITEM_TYPE_CODE	OPTION_FLAG
1	Model (top level)	1			Model	
2	.Class A	2	1	1	Option Class	Y
3	..Class A Option	3	1	2	Standard	Y
2	.Model B	4	1	1	Model	Y
3	..Model B Option	5	1	4	Standard	Y
3	..Class C	6	1	4	Option Class	Y
4	...Class C Option	7	1	6	Kit	Y

Table 6 – 9 (Page 1 of 1)



Configure to Order
(*Oracle Order Entry Reference Manual*)

Bills of Material
(*Oracle Bills of Material Reference Manual*)

Importing Schedule Details

You can import schedule details for the item on an order line. When you import a schedule detail, you must specify the line and you can only specify details for the same inventory item that is on the order line.

Importing Customer Requested Shipment Schedules

Imported customer requested shipment schedules must be separate order lines, unlike online order entry, where shipment schedules are related to an order line. If you are creating shipment schedules to gain price advantages, you will need to import the appropriate price for each line. If you use automatic pricing, each order line (imported shipment schedule) is evaluated separately and volume discounts across lines are not considered.



Scheduling Orders
(*Oracle Order Entry Reference Manual*)

Managing Customer Requests to Ship Orders Complete

OrderImport provides a line level field to control whether the lines on a order can ship independently or must ship together. This field, SHIP_SET_NUMBER, if populated with the same number for each line will require the entire order to be ready before it will be released for picking.

Using MultiCurrency and Price Lists

You must indicate the currency of each order you import. If you have price lists in different currencies you can load both the price list and matching currency through OrderImport. Or you can default the price list according to the standard value rules you have defined. If the defaulted price list does not have the same currency as what you imported, Oracle Order Entry rejects the transaction.



Using Standard Value Rule Sets
(*Oracle Order Entry Reference Manual*)



Define Standard Value Rule Sets
(*Oracle Order Entry Reference Manual*)

Changes to Imported Orders

You can import changes to imported orders. When entering changes to orders you need to identify the order, order line(s), price adjustments and sales credits, and match the columns in the SO_HEADERS table to the corresponding ones in the SO_HEADERS_INTERFACE table, shown in the following table.

Specific Record to Change	Match This Order Entry Column...	...to This OrderImport Interface Column
Header	SO_HEADERS columns	SO_HEADERS_INTERFACE columns
	ORIGINAL_SYSTEM_SOURCE_CODE	ORDER_SOURCE_ID
	ORIGINAL_SYSTEM_REFERENCE	ORIGINAL_SYSTEM_REFERENCE
Line	SO_HEADERS columns	SO_LINES_INTERFACE columns
	ORIGINAL_SYSTEM_SOURCE_CODE	ORDER_SOURCE_ID
	ORIGINAL_SYSTEM_REFERENCE	ORIGINAL_SYSTEM_REFERENCE
	SO_LINES column	SO_LINES_INTERFACE column
	ORIGINAL_SYSTEM_LINE_REFERENCE	ORIGINAL_SYSTEM_LINE_REFERENCE
Price Adjustments	SO_HEADERS columns	SO_PRICE_ADJUSTMENTS_INTERFACE columns
	ORIGINAL_SYSTEM_SOURCE_CODE	ORDER_SOURCE_ID
	ORIGINAL_SYSTEM_REFERENCE	ORIGINAL_SYSTEM_REFERENCE
	SO_LINES column	SO_PRICE_ADJUSTMENTS_INTERFACE column
	ORIGINAL_SYSTEM_LINE_REFERENCE	ORIGINAL_SYSTEM_LINE_REFERENCE
Sales Credits	SO_HEADERS columns	SO_SALES_CREDITS_INTERFACE columns
	ORIGINAL_SYSTEM_SOURCE_CODE	ORDER_SOURCE_ID
	ORIGINAL_SYSTEM_REFERENCE	ORIGINAL_SYSTEM_REFERENCE
	SO_LINES column	SO_SALES_CREDITS_INTERFACE column
	ORIGINAL_SYSTEM_LINE_REFERENCE	ORIGINAL_SYSTEM_LINE_REFERENCE

Table 6 – 10 (Page 1 of 1)

You need to specify the operation code to determine the kind of change you want to import. Use the OPERATION_CODE column in the interface tables (except for SO_LINE_DETAILS_INTERFACE) to indicate whether you are adding, modifying or deleting imported order information.

When you want to create a new imported order, enter INSERT or null in SO_HEADERS_INTERFACE.OPERATION_CODE. When you want to delete an

imported order, enter DELETE in SO_HEADERS_INTERFACE. OPERATION_CODE.

When you want to update an order, you must reenter all the required and conditionally required columns for the order along with any changes. When you create new order lines, order price adjustments, order line price adjustments, order sales credit, or order line sales credits for an order you have already imported, you must update the order header, and for line level additions, the order line.

When you want to create or delete an order line, order price adjustment or order sales credit for an order you have already imported, enter UPDATE in SO_HEADERS_INTERFACE. OPERATION_CODE (and reenter all the required columns), and INSERT or DELETE in the appropriate OPERATION_CODE column of any of SO_LINES_INTERFACE, SO_PRICE_ADJUSTMENTS_INTERFACE, or SO_SALES_CREDITS_INTERFACE.

When you want to create or delete an order line price adjustment or order line sales credit for an order line you have already imported, you must enter UPDATE in SO_HEADERS_INTERFACE. OPERATION_CODE (and reenter all the required columns), UPDATE in SO_LINES_INTERFACE. OPERATION_CODE (and reenter all the required columns), and INSERT or DELETE in the appropriate OPERATION_CODE column of SO_PRICE_ADJUSTMENTS_INTERFACE, or SO_SALES_CREDITS_INTERFACE.

When you want to delete an order line for a model, enter UPDATE in SO_HEADERS_INTERFACE. OPERATION_CODE (and reenter all the required columns), and DELETE in the appropriate OPERATION_CODE column of SO_LINES_INTERFACE for the model. This action deletes both the model line and all its associated option classes, option items, and included items.

When you want to update an order line for a model that you have already imported, enter UPDATE in SO_HEADERS_INTERFACE. OPERATION_CODE (and reenter all the required columns), and DELETE and INSERT in the appropriate OPERATION_CODE column of any of SO_LINES_INTERFACE, SO_PRICE_ADJUSTMENTS_INTERFACE, or SO_SALES_CREDITS_INTERFACE for the model and all its associated option classes, option items, and included items. You must give the inserting model a new line number since you are simultaneously deleting and inserting in order to update the line.

When you want to update an order that has multiple order lines, some reserved or demanded and some not, you can update the unscheduled order line(s) without affecting the schedule status of the reserved or demanded lines. Enter UPDATE in SO_HEADERS_INTERFACE.

OPERATION_CODE (and reenter all the required columns), and set SO_LINES_INTERFACE.OPERATION_CODE to UPDATE for the unscheduled order lines you want to modify. You cannot import changes to an order line that is reserved or demanded. Instead, query the order in the Enter Orders form or the Schedule Orders form and modify the corresponding field(s).

You can insert a new order line into an existing schedule group, (Ship Set, ATO Configuration, or PTO Ship Complete Configuration), up until the schedule group is demanded or reserved.

When you want to update an order line that is part of a schedule group, (Ship Set, ATO Configuration, or PTO Ship Complete Configuration), you can modify Ordered Quantity and Schedule Date prior to demanding or reserving the schedule group. The following columns cannot be changed at any time, through OrderImport, for lines in a schedule group, Warehouse, Ship to Location, Freight Carrier, Shipment Priority, and Demand Class. Instead, query the order in the Enter Orders form or the Schedule Orders form and modify the schedule group attributes.

When you import changes to any schedule details (SO_LINE_DETAILS_INTERFACE) you need to reimport *all* the schedule details for the order line. Enter UPDATE in SO_HEADERS_INTERFACE.

OPERATION_CODE (and reenter all the required columns), and set SO_LINES_INTERFACE.OPERATION_CODE to UPDATE for the corresponding order line. OrderImport deletes existing schedule details for the order line and replaces those with the newly imported schedule detail information. The new schedule details totals must add up to the total quantity for the order line. If an order line is reserved or demanded, any changes to schedule details through OrderImport will result in the order line and its schedule details being unscheduled. You can reschedule the entire *order*, when changing schedule details, by setting SO_HEADERS_INTERFACE.SCHEDULE_STATUS_CODE to RESERVED or DEMANDED.



Attention: If you want to delete a column, you cannot do so by updating the interface table. You should query the order in the Enter Orders form and delete the data from the corresponding field.

The operation codes that you need to indicate for each kind of changes are outlined in the following table. Operation codes in angle brackets (< >) are optional; all others are required for the operation.

Operation	SO_HEADERS_INTERFACE	SO_LINES_INTERFACE	SO_PRICE_ADJUSTMENTS_INTERFACE	SO_SALES_CREDITS_INTERFACE
Add order	<Insert>	<Insert>	<Insert>	<Insert>
Add order line	Update	Insert	<Insert>	<Insert>
Delete order	Delete	Delete	<Delete>	Delete
Delete order line	Update	Delete	<Delete>	<Delete>
Update order	Update	--	--	--
Update order line	Update	Update	<Update>	<Update>
Update order price adjustment	Update	--	Update	--
Update order sales credit	Update	--	--	Update
Update line price adjustment	Update	Update	Update	--
Update line sales credit	Update	Update	--	Update
Delete model	Update	Delete		
Update model	Update	Delete and Insert	<Insert>	<Insert>

Table 6 – 11 (Page 1 of 1)

Update Statements

The following update statements are necessary to reprocess orders in the OrderImport interface tables. First fix the problems/errors in the orders and ensure they are ready for OrderImport to process them again. Then use the following statement in SQL*Plus for each interface table to remove values from the necessary columns.



Attention: Values or IDs specific to your situation appear in *italics* within angle brackets (< >). Be prepared to substitute the specific value or ID for the generic value in the angle brackets. Do not include the angle brackets when substituting your values for these names.

```
update so_headers_interface
set request_id = NULL, error_flag = NULL,
interface_status = NULL, report_summary = NULL
where
```

<insert where clause here>

The where clause could include statements like:

- where request_id = <xxxxxx>
- where order_source_id = <yyyy>
- where original_system_reference in
('<aaa>', '<bbb>')
- any combination of the above

Using IDs

You can define sources and determine whether to use IDs in the Define OrderImport Sources form. If you elect to use IDs, you must populate all required ID columns in each table. You may also populate the associated name columns, however, Oracle Order Entry does not consider the names when you elect to use IDs.

If you elect to *not* use IDs, then you must populate all the required name columns in each table. You may also populate the ID columns, however, except for those columns listed below, Oracle Order Entry does not consider the IDs when you elect to use names.

If you are importing using names and you have a nonunique customer, address information, contact, or salesperson, you can include the IDs for only the columns indicated in the following table. OrderImport uses the IDs along with the names to identify the specific customer, address or salesperson. The following table shows the corresponding name and ID columns.

NAME COLUMN	ID COLUMN
SO_HEADERS_INTERFACE	
CUSTOMER_NAME CUSTOMER_NUMBER	CUSTOMER_ID
ORDERED_BY_CONTACT_FIRST_NAME ORDERED_BY_CONTACT_LAST_NAME	ORDERED_BY_CONTACT_ID
INVOICE_CUSTOMER (name)	INVOICE_CUSTOMER_ID

Table 6 – 12 (Page 1 of 2)

NAME COLUMN		ID COLUMN
INVOICE_ADDRESS1 INVOICE_ADDRESS2 INVOICE_ADDRESS3 INVOICE_ADDRESS4 INVOICE_CITY INVOICE_STATE INVOICE_COUNTY INVOICE_COUNTRY INVOICE_POSTAL_CODE	INVOICE_ADDRESS_ID	
INVOICE_TO_CONTACT_FIRST_NAME INVOICE_TO_CONTACT_LAST_NAME	INVOICE_TO_CONTACT_ID	
SHIP_TO_CUSTOMER (name)	SHIP_TO_CUSTOMER_ID	
SHIP_ADDRESS1 SHIP_ADDRESS2 SHIP_ADDRESS3 SHIP_ADDRESS4 SHIP_CITY SHIP_STATE SHIP_COUNTY SHIP_COUNTRY SHIP_POSTAL_CODE	SHIP_ADDRESS_ID	
SHIP_TO_CONTACT_FIRST_NAME SHIP_TO_CONTACT_LAST_NAME	SHIP_TO_CONTACT_ID	
SALESREP_NAME	SALESREP_ID	
SALESREP_NUMBER		
SO_LINES_INTERFACE		
SHIP_TO_CUSTOMER (name)	SHIP_TO_CUSTOMER_ID	
SHIP_ADDRESS1 SHIP_ADDRESS2 SHIP_ADDRESS3 SHIP_ADDRESS4 SHIP_CITY SHIP_STATE SHIP_COUNTY SHIP_COUNTRY SHIP_POSTAL_CODE	SHIP_ADDRESS_ID	
SHIP_TO_CONTACT_FIRST_NAME SHIP_TO_CONTACT_LAST_NAME	SHIP_TO_CONTACT_ID	

Table 6 - 12 (Page 2 of 2)

If the name, or number, and ID combination you supply does not match an existing customer, address or salesperson information, OrderImport will reject the records. (For the INVOICE_CUSTOMER

and SHIP_TO_CUSTOMER columns, you must enter the name, you cannot use a customer number.)



Define OrderImport Sources
(*Oracle Order Entry Reference Manual*)

Standard Value Rule Sets

In addition to entering data in the columns of the interface tables, you can utilize the standard value defaulting feature to fill in values to each imported order. You specify a standard value rule set with each order type. Any information that you enter in the columns of the interface tables, which can also be defaulted by the standard value rule set, will be treated as values that a user entered, so you should set the override fields of the Define Standard Value Rule Sets form according to how you want Oracle Order Entry to prioritize user entered values versus defaulted values.



Using Standard Value Rule Sets
(*Oracle Order Entry Reference Manual*)



Define Standard Value Rule Sets
(*Oracle Order Entry Reference Manual*)

OrderImport Validation

OrderImport validates your data for compatibility with Oracle Order Entry. OrderImport validates your data by ensuring that the columns in Oracle Order Entry OrderImport interface tables reference the appropriate and active values and columns in Oracle Order Entry. Whenever IDs are used, Oracle Order Entry OrderImport automatically validates the name and ID combination, as well as the individual name and ID.

If you are importing open orders, then for those attributes that must be valid, the following rules apply: if an attribute has start and end dates (such as agreements or price lists) OrderImport validates that the end date is greater than the current date; if the attribute has an enable flag (such as QuickCodes) OrderImport validates that the flag is set to enabled; if the attribute has a status (such as customer) OrderImport validates that the status is Active.

If you are importing closed orders, then for those attributes that must just exist, the start and end dates, enable flag, and status are irrelevant; the attribute only needs to exist in your database.

Transaction Level Validation

OrderImport validates the following attributes to ensure that your transactions contain the appropriate information for Oracle Order Entry:

- valid customer
- valid ship to and bill to locations/addresses
- valid customer agreement
- valid contacts
- valid order type
- valid OrderImport source
- valid order category
- valid entry status
- valid salesperson
- valid price list
- valid currency
 - valid conversion type
- valid payment terms
- valid payment type
 - valid credit card type
- valid sales channel
- valid shipment priority, FOB, freight carrier and freight terms
- valid accounting rule
- valid invoicing rule

Transaction Line Level Validation

OrderImport validates the following attributes to ensure that your transaction lines contain the appropriate information for Oracle Order Entry:

- valid unit of measure
- valid inventory item
- valid ship to address
- valid freight carrier

- valid discount (price adjustment)
- valid warehouse
- valid configuration (against bill of material for the top model)

Complete Orders Validation

Since you should only be importing complete orders for historical purposes, OrderImport relaxes validation on this type of order. Instead of requiring that attributes be valid as far as having an active status, having a valid start and end date range, or being enable, they only need to exist in your database. And in the case of an inventory item, the Customer Ordered Item attribute *must be* set to Yes, and the Customer Orders Enabled attribute can be set to either Yes or No.

- existing customer
- existing ship to and bill to addresses
- existing and orderable inventory item
- existing price list
- existing salesperson

Security Rules and Standard Value Rule Sets Validation

OrderImport checks security rule and standard value rule sets after every order is inserted or updated in the interface tables. If you try to update an order at a point in its order cycle that violates your security rules, OrderImport gives you an error. Standard value rule sets are reevaluated after every update to see if any defaulted values should be changed according to your rule.

Sales Credit Validation

OrderImport validates the following attributes to ensure that your sales credit information contains the appropriate information for Oracle Order Entry:

- valid sales credit type
- valid salesperson
- total revenue percent for the order or each line is not greater than 100

Price Adjustment Validation

OrderImport validates the following attributes to ensure that your price adjustments contain the appropriate information for Oracle Order Entry:

- valid discount (price adjustment)
- valid price list

Schedule Details Validation

OrderImport validates the following attributes to ensure that your schedule details contain the appropriate information for Oracle Order Entry:

- valid subinventory
- valid lot
- valid revision
- total quantity of schedule details equals total ordered quantity on associated order line

Required Fields for Booked Orders

If you are importing booked orders, verify that you have either entered, or your standard value rules will default all the fields required to book the order. If all the required fields are not completed, OrderImport will reject the records. At the beginning of each table description we provide tables that indicate conditionally required for booking columns. For more information on required for booking fields refer to the Required Fields for Entering Orders Appendix in the *Oracle Order Entry Reference Manual*.



Required Fields for Entering Orders Appendix
(*Oracle Order Entry Reference Manual*)

Running OrderImport

Running OrderImport can be a one step process, depending on the accuracy of your import program. If your import program converts your transaction information from other sources into the required format, and all your data passes the validation in Oracle Order Entry, then you should be able to successfully run OrderImport in one execution. However, if you load data into the OrderImport interface

tables that Oracle Order Entry validation rejects, then OrderImport informs you of the specific validation errors in the concurrent request report. In this case, you need to correct any errors and reimport the corrected data. Before you reimport the corrected data, you need to set the request ID in all five interface tables to null, otherwise OrderImport will not process any records. Run OrderImport using the Run Concurrent Programs form in Oracle Order Entry. You can import one or multiple sources of data at one time, based on the parameters you set for the OrderImport program from this form.



Run Concurrent Programs
(Oracle Order Entry Reference Manual)

Oracle Order Entry Interface Tables and Column Descriptions

OrderImport uses the following tables and columns:

SO_HEADERS_INTERFACE

The following graphic summarizes the SO_HEADERS_INTERFACE table. Database columns with a check under Conditionally Required for Booking are optional if you are importing an order with an Entry Status of **Entered**. If you are importing an order with an Entry Status of **Booked**, you must enter the ID column if you are using IDs, or the name column if you are not using IDs. Refer to the Using IDs section on page 6 – 66, and the Required Fields for Booked Orders section on page 6 – 71 for more details.

Database columns not included in this table are for internal use only. Each column is described in detail below starting on page 6 – 78.

SO_HEADERS_INTERFACE Column Name	Type	Required (C indicates Condi- tionally Required)	Condi- tionally Required for Booking	Derived	Optional
CREATION_DATE	Date	✓			
CREATED_BY	Number	✓			

Table 6 – 13 SO_HEADERS_INTERFACE Table (Page 1 of 7)

SO_HEADERS_ INTERFACE Column Name	Type	Required (C indicates Condi- tionally Required)	Condi- tionally Required for Booking	Derived	Optional
LAST_UPDATE_DATE	Date	✓			
LAST_UPDATED_BY	Number	✓			
LAST_UPDATE_LOGIN	Number				✓
PROGRAM_ APPLICATION_ID	Number				✓
PROGRAM_ID	Number				✓
PROGRAM_UPDATE_DATE	Date				✓
REQUEST_ID	Number				✓
ORIGINAL_SYSTEM_ REFERENCE	Var- char2(50)	✓			
ORDER_NUMBER_ SOURCE_ID	Number			✓	
CUSTOMER_NAME	Var- char2(50)		✓		
CUSTOMER_NUMBER	Var- char2(30)		✓		
ORDER_TYPE	Var- char2(30)	C			
ORDER_TYPE_ID	Number	C			
ORDER_SOURCE_ID	Number	✓			
ORDER_CATEGORY	Var- char2(30)	✓			
DATE_ORDERED	Date	✓			
CURRENCY_CODE	Var- char2(15)	✓			
CONVERSION_RATE	Number	C			
CONVERSION_DATE	Date	C			
CONVERSION_TYPE_CODE	Var- char2(30)	C			

Table 6 – 13 SO_HEADERS_INTERFACE Table (Page 2 of 7)

SO_HEADERS_ INTERFACE Column Name	Type	Required (C indicates Condi- tionally Required)	Condi- tionally Required for Booking	Derived	Optional
SALESREP_NAME	Var- char2(30)		✓		
SALESREP_ID	Number		✓		
INVOICE_CUSTOMER	Var- char2(50)		✓		
INVOICE_CUSTOMER_ID	Number		✓		
INVOICE_TO_SITE_USE_ID	Number			✓	
INVOICE_ADDRESS_ID	Number		✓		
INVOICE_ADDRESS1 – 4	Var- char2(240)		✓		
INVOICE_CITY	Var- char2(50)		✓		
INVOICE_STATE	Var- char2(50)		✓		
INVOICE_COUNTY	Var- char2(50)		✓		
INVOICE_COUNTRY	Var- char2(50)		✓		
INVOICE_POSTAL_CODE	Var- char2(30)		✓		
SHIP_TO_SITE_USE_ID	Number			✓	
SHIP_TO_CUSTOMER	Var- char2(50)		✓		
SHIP_TO_CUSTOMER_ID	Number		✓		
SHIP_ADDRESS_ID	Number		✓		
SHIP_ADDRESS1 – 4	Var- char2(240)		✓		
SHIP_CITY	Var- char2(50)		✓		

Table 6 – 13 SO_HEADERS_INTERFACE Table (Page 3 of 7)

SO_HEADERS_ INTERFACE Column Name	Type	Required (C indicates Condi- tionally Required)	Condi- tionally Required for Booking	Derived	Optional
SHIP_STATE	Var- char2(50)		✓		
SHIP_COUNTY	Var- char2(50)		✓		
SHIP_COUNTRY	Var- char2(50)		✓		
SHIP_POSTAL_CODE	Var- char2(30)		✓		
PRICE_LIST_NAME	Var- char2(30)		✓		
PRICE_LIST_ID	Number		✓		
TERMS_NAME	Var- char2(15)		✓		
TERMS_ID	Number		✓		
CYCLE_ID	Number			✓	
ENTERED_STATE_NAME	Var- char2(30)	C			
ENTERED_STATE_ID	Number	C			
ENTERED_STATE_DATE	Date				✓
COMPLETE_FLAG	Varchar2(1)				✓
AGREEMENT_NAME	Var- char2(30)				✓

Table 6 – 13 SO_HEADERS_INTERFACE Table (Page 4 of 7)

SO_HEADERS_ INTERFACE Column Name	Type	Required (C indicates Condi- tionally Required)	Condi- tionally Required for Booking	Derived	Optional
AGREEMENT_ID	Number				✓
DATE_ REQUESTED_CURRENT	Date				✓
ORDERED_BY_ CONTACT_FIRST_NAME	Var- char2(30)				✓
ORDERED_BY_ CONTACT_LAST_NAME	Var- char2(50)				✓
ORDERED_BY_ CONTACT_ID	Number				✓
SALES_ CHANNEL_CODE	Var- char2(30)				✓
PURCHASE_ORDER_NUM	Var- char2(50)				✓
INVOICE_TO_ CONTACT_FIRST_NAME	Var- char2(30)				✓
INVOICE_TO_ CONTACT_LAST_NAME	Var- char2(50)				✓
INVOICE_TO_ CONTACT_ID	Number				✓
SHIP_TO_ CONTACT_FIRST_NAME	Var- char2(30)				✓
SHIP_TO_ CONTACT_LAST_NAME	Var- char2(50)				✓
SHIP_TO_ CONTACT_ID	Number				✓
SHIPMENT_ PRIORITY_CODE	Var- char2(30)				✓
SHIP_METHOD_CODE	Var- char2(30)				✓
FREIGHT_TERMS_CODE	Var- char2(30)				✓

Table 6 – 13 SO_HEADERS_INTERFACE Table (Page 5 of 7)

SO_HEADERS_INTERFACE Column Name	Type	Required (C indicates Condi- tionally Required)	Condi- tionally Required for Booking	Derived	Optional
FOB_CODE	Var- char2(30)				✓
SHIPPING_INSTRUCTIONS	Var- char2(240)				✓
PACKING_INSTRUCTIONS	Var- char2(240)				✓
DATE_SHIPPED	Date				✓
PAYMENT_TYPE_CODE	Var- char2(30)				✓
PAYMENT_AMOUNT	Number				✓
CHECK_NUMBER	Var- char2(50)				✓
CREDIT_CARD_CODE	Var- char2(30)				✓
CREDIT_CARD_HOLDER_NAME	Var- char2(50)				✓
CREDIT_CARD_NUMBER	Var- char2(50)				✓
CREDIT_CARD_EXPIRATION_DATE	Date				✓
CREDIT_CARD_APPROVAL_CODE	Var- char2(50)				✓
ACCOUNTING_RULE	Var- char2(30)		✓		
ACCOUNTING_RULE_ID	Number		✓		
INVOICING_RULE	Var- char2(30)		✓		
INVOICING_RULE_ID	Number		✓		
CONTEXT	Var- char2(30)				✓

Table 6 – 13 SO_HEADERS_INTERFACE Table (Page 6 of 7)

SO_HEADERS_ INTERFACE Column Name	Type	Required (C indicates Condi- tionally Required)	Condi- tionally Required for Booking	Derived	Optional
ATTRIBUTE1 - 15	Var- char2(150)				✓
SCHEDULE_STATUS_CODE	Var- char2(30)				✓
OPERATION_CODE	Var- char2(30)				✓
SALESREP_NUMBER	Number		✓		✓
TAX_EXEMPT_FLAG	Varchar2(1)				✓
TAX_EXEMPT_NUM	Var- char2(30)				✓
TAX_EXEMPT_ REASON_CODE	Var- char2(30)	C			
APPLY_STANDARD _NOTES	Varchar2(1)				✓
ORG_ID	Number				✓

Table 6 – 13 SO_HEADERS_INTERFACE Table (Page 7 of 7)

CREATION_DATE Not Null **DATE.**

Enter the date that you originally entered the data into your feeder system.

Validation: Standard Date Validation

Destination: SO_HEADERS.CREATION_DATE

CREATED_BY Not Null **NUMBER**

Enter an identification number that you can use to identify the user who created the record.

Validation: None

Destination: SO_HEADERS.CREATED_BY

LAST_UPDATE_DATE	Not Null	DATE
-------------------------	----------	-------------

Enter the current date that you are entering data into your feeder system.

Validation: Standard Date Validation

Destination: SO_HEADERS.LAST_UPDATE_DATE

LAST_UPDATED_BY	Not Null	NUMBER
------------------------	----------	---------------

Enter an identification number that you can use to identify the user who created or who most recently modified the record.

Validation: None

Destination: SO_HEADERS.LAST_UPDATED_BY

LAST_UPDATE_LOGIN		NUMBER
--------------------------	--	---------------

Enter an identification number that you can use to identify the feeder system that is supplying Oracle Order Entry with this data.

Validation: None

Destination: SO_HEADERS.LAST_UPDATE_LOGIN

PROGRAM_APPLICATION_ID	NUMBER(15)
-------------------------------	-------------------

This column is for internal use only.

PROGRAM_ID	NUMBER(15)
-------------------	-------------------

This column is for internal use only.

PROGRAM_UPDATE_DATE	DATE
----------------------------	-------------

This column is for internal use only.

REQUEST_ID	NUMBER(15)
-------------------	-------------------

This column is for internal use only. It is populated with the concurrent manager request ID for the run of the interface each time you submit the OrderImport program. In order for records to be processed, their request ID must be null or equal to the request ID being processed.

ORIGINAL_SYSTEM_		
REFERENCE	Not Null	VARCHAR2(50)

Enter the order number or ID from your original feeder system. The order number of ID you enter provides you with an audit trail from OrderImport to your feeder system.

Validation: If the value you enter already exists, then the OPERATION_CODE must be **UPDATE** or **DELETE**.

Destination: SO_HEADERS.ORIGINAL_SYSTEM_REFERENCE
(This order number appears on the Enter Orders and View Orders forms.)

ORDER_NUMBER_	
SOURCE_ID	NUMBER

The information for this column is derived from the order type and used to generate the order number. It contains the ID of the order number source to be used when generating the order number from SO_ORDER_NUMBER_SOURCES.

Validation: None

Destination: None



Define Order Type
Define Order Number Sources
(Oracle Order Entry Reference Manual)

CUSTOMER_NAME	VARCHAR2(50)
----------------------	---------------------

Enter the full name of the customer for this order. If you are using IDs, you must enter the CUSTOMER_ID. If you are not using IDs, then you must enter the CUSTOMER_NAME or CUSTOMER_NUMBER, and if your CUSTOMER_NAME and/or CUSTOMER_NUMBER is not unique, you must also enter the CUSTOMER_ID.

Validation: RA_CUSTOMERS.CUSTOMER_NAME

Destination: inserts RA_CUSTOMERS.CUSTOMER_ID into
SO_HEADERS.CUSTOMER_ID where
SO_HEADERS_INTERFACE.CUSTOMER_NAME
= RA_CUSTOMERS.CUSTOMER_NAME

CUSTOMER_NUMBER	VARCHAR2(30)
------------------------	---------------------

Enter the customer number for this order. If you are using IDs, you must enter the CUSTOMER_ID. If you are not using IDs, then you must enter the CUSTOMER_NAME or CUSTOMER_NUMBER, and if your CUSTOMER_NAME and/or CUSTOMER_NUMBER is not unique, you must also enter the CUSTOMER_ID.

Validation: RA_CUSTOMERS.CUSTOMER_NUMBER

Destination: None

CUSTOMER_ID	NUMBER
--------------------	---------------

Enter the customer ID that matches your customer name. If you are using IDs, you must enter the CUSTOMER_ID. If you are not using IDs, then you must enter the CUSTOMER_NAME or CUSTOMER_NUMBER, and if your CUSTOMER_NAME and/or CUSTOMER_NUMBER is not unique, you must also enter the CUSTOMER_ID.

Validation: RA_CUSTOMERS.CUSTOMER_ID

Destination: SO_HEADERS.CUSTOMER_ID

ORDER_TYPE	Conditionally Not Null	VARCHAR2(30)
-------------------	------------------------	---------------------

Enter the name of the order type to assign to your order. If you are using IDs, you must enter the ORDER_TYPE_ID. If you are not using IDs, then you must enter the ORDER_TYPE_NAME.

Validation: SO_ORDER_TYPES.NAME

Destination: inserts SO_ORDER_TYPES.ORDER_TYPE_ID into SO_HEADERS.ORDER_TYPE_ID where SO_HEADERS_INTERFACE.ORDER_TYPE = SO_ORDER_TYPES.NAME

ORDER_TYPE_ID	Conditionally Not Null	NUMBER
----------------------	------------------------	---------------

Enter the order type ID that matches your order type. If you are using IDs, you must enter the ORDER_TYPE_ID. If you are not using IDs, then you must enter the ORDER_TYPE_NAME.

Validation: SO_ORDER_TYPES.ORDER_TYPE_ID

Destination: SO_HEADERS.ORDER_TYPE_ID

ORDER_SOURCE_ID	Not Null	NUMBER
------------------------	----------	---------------

Enter the order import source ID. This tells OrderImport whether to use IDs or the name columns, and is used to determine which records in the interface table to process.

Validation: SO_ORDER_SOURCES.ORDER_SOURCE_ID

Destination: None



Define OrderImport Sources
(Oracle Order Entry Reference Manual)

ORDER_CATEGORY	Not Null	VARCHAR2(30)
-----------------------	----------	---------------------

Enter **R** in this column. This stands for a Regular order.

Validation: Must equal **R**

Destination: SO_HEADERS.ORDER_CATEGORY

DATE_ORDERED	Not Null	DATE
---------------------	----------	-------------

Enter the date on which this order was placed in the original system.

Validation: Standard Date Validation

Destination: SO_HEADERS.DATE_ORDERED

CURRENCY_CODE	Not Null	VARCHAR2(15)
----------------------	----------	---------------------

Enter the currency for the order.

Validation: FND_CURRENCIES

Destination: SO_HEADERS.CURRENCY_CODE

CONVERSION_RATE	Conditionally Not Null	NUMBER
------------------------	------------------------	---------------

Enter the currency conversion rate for the order if you entered a currency code other than the functional currency for your set of books and a conversion_type_code equal to **User**.

Validation: None

Destination: SO_HEADERS.CONVERSION_RATE

CONVERSION_DATE	Conditionally Not Null	DATE
------------------------	------------------------	-------------

Enter the currency conversion date for which the conversion rate is valid for the order, if you entered a currency code other than the

functional currency for your set of books and a conversion_type_code equal to **User**.

Validation: None

Destination: SO_HEADERS.CONVERSION_DATE

CONVERSION_ TYPE_CODE	Conditionally Not Null	VARCHAR2(30)
--	------------------------	---------------------

Enter the currency conversion type for the order if you entered a currency code other than the functional currency for your set of books.

Validation: GL_DAILY_CONVERSION_TYPES.
CONVERSION_TYPE

Destination: SO_HEADERS.CONVERSION_TYPE_CODE

SALESREP_NAME	VARCHAR2(30)
----------------------	---------------------

Enter the name of the primary salesperson for this order. (If you do not enter explicit sales credit records, OrderImport will automatically assign 100% quota credit for the whole order to this salesperson). Refer to Table 6 – 12 on page 6 – 67 in the Using IDs section above for more details on this conditionally required for booking column.

Validation: RA_SALESREPS.NAME

Destinations: inserts RA_SALESREPS.SALESREP_ID INTO
SO_HEADERS.SALESREP_ID where
RA_SALESREP.NAME =
SO_HEADERS_INTERFACE.SALESREP_NAME

SALESREP_ID	NUMBER
--------------------	---------------

Enter the salesrep id for the salesperson. Refer to Table 6 – 12 on page 6 – 67 in the Using IDs section above for more details on this conditionally required for booking column.

Validation: RA_SALESREPS.SALESREP_ID

Destination: SO_HEADERS.SALESREP_ID

INVOICE_CUSTOMER	VARCHAR2(50)
-------------------------	---------------------

Enter the name of the customer to receive the invoice for this order. Refer to Table 6 – 12 on page 6 – 67 in the Using IDs section above for more details on this conditionally required for booking column.

Validation: Validates whether the INVOICE_CUSTOMER exists in RA_CUSTOMERS and RA_CUSTOMERS.STATUS is **Active**. (If the customer name is not unique, you must enter an ID in the INVOICE_CUSTOMER_ID column.) Also, if **OE: Customer Relationships** is set to Yes, then the Invoice To Location must be the same as the ordering customer or a related customer.

Destination: None

INVOICE_CUSTOMER_ID	Conditionally Not Null	NUMBER
---------------------	------------------------	--------

Enter the customer ID of your INVOICE_CUSTOMER. Refer to Table 6 – 12 on page 6 – 67 in the Using IDs section above for more details on this conditionally required for booking column.

Validation: Validates whether the INVOICE_CUSTOMER_ID exists in RA_CUSTOMERS and RA_CUSTOMERS.STATUS is **Active**. Also, if **OE: Customer Relationships** is set to Yes, then the Invoice To Location must be the same as the ordering customer or a related customer.

Destination: None

INVOICE_TO_SITE_USE_ID	NUMBER
------------------------	--------

Leave this column null, it is derived from the INVOICE_ADDRESS_ID column.

Destination: inserts RA_SITE_USE.SITE_USE_ID into SO_HEADERS.INVOICE_SITE_USE_ID where RA_ADDRESSES.ADDRESS_ID = RA_SITE_USES.ADDRESS_ID and RA_SITE_USES.SITE_USE_CODE = 'BILL_TO'

INVOICE_ADDRESS_ID	NUMBER
--------------------	--------

Enter the ADDRESS_ID which matches your complete invoice address location. This value is used to derive SO_LINES_INTERFACE.INVOICE_TO_SITE_USE_ID. Refer to Table 6 – 12 on page 6 – 67 in the Using IDs section above for more details on this conditionally required for booking column.

Validation: Must represent a valid location for the INVOICE_CUSTOMER and exist in RA_SITE_USES where SITE_USE_CODE is **BILL_TO**. Validates whether the address is valid and exists in RA_ADDRESSES; must be a valid address for the INVOICE_CUSTOMER. Also, if **OE: Customer Relationships** is set to Yes, then the location must be a valid site use for the Invoice To customer and be the same as the ordering customer or a related customer.

INVOICE_
ADDRESS1 – 4 **VARCHAR2(240)**

Enter the street address (4 lines) for your invoice address. Refer to Table 6 – 12 on page 6 – 67 in the Using IDs section above for more details on this conditionally required for booking column.

INVOICE_CITY **VARCHAR2(50)**

Enter the city for your invoice address. Refer to Table 6 – 12 on page 6 – 67 in the Using IDs section above for more details on this conditionally required for booking column.

INVOICE_STATE **VARCHAR2(50)**

Enter the state or province for your invoice address. Refer to Table 6 – 12 on page 6 – 67 in the Using IDs section above for more details on this conditionally required for booking column.

INVOICE_COUNTY **VARCHAR2(50)**

Enter the county for your invoice address. Refer to Table 6 – 12 on page 6 – 67 in the Using IDs section above for more details on this conditionally required for booking column.

INVOICE_COUNTRY **VARCHAR2(50)**

Enter the country for your invoice address. Refer to Table 6 – 12 on page 6 – 67 in the Using IDs section above for more details on this conditionally required for booking column.

Validation: FND_TERRITORIES

INVOICE_	
POSTAL_CODE	VARCHAR2(30)

Enter the postal code for your invoice address. Refer to Table 6 – 12 on page 6 – 67 in the Using IDs section above for more details on this conditionally required for booking column.

SHIP_TO_SITE_USE_ID	NUMBER
----------------------------	---------------

Leave this column null, it is derived from the SHIP_ADDRESS_ID column.

Destination: inserts RA_SITE_USE.SITE_USE_ID into SO_HEADERS.SHIP_TO_SITE_USE_ID where RA_ADDRESSES.ADDRESS_ID = RA_SITE_USES.ADDRESS_ID and RA_SITE_USES.SITE_USE_CODE = 'SHIP_TO'

SHIP_TO_CUSTOMER	VARCHAR2(50)
-------------------------	---------------------

Enter the name of the customer to receive the shipment. Refer to Table 6 – 12 on page 6 – 67 in the Using IDs section above for more details on this conditionally required for booking column.

Validation: Validates whether the SHIP_TO_CUSTOMER exists in RA_CUSTOMERS and RA_CUSTOMERS.STATUS is **Active**. (If the customer name is not unique, you must enter an ID in the SHIP_TO_CUSTOMER_ID column.) Also, if **OE: Customer Relationships** is set to Yes, then the location must be the same as the ordering customer or a related customer.

Destination: None

SHIP_TO_		
CUSTOMER_ID	Conditionally Not Null	NUMBER

Enter the customer ID of your Ship To Customer. If you are using IDs, you must enter the SHIP_TO_CUSTOMER_ID. If you are not using IDs, then you must enter the SHIP_TO_CUSTOMER. Refer to Table 6 – 12 on page 6 – 67 in the Using IDs section above for more details on this conditionally required for booking column.

Validation: Validates whether the SHIP_TO_CUSTOMER_ID exists in RA_CUSTOMERS and RA_CUSTOMERS.STATUS is **Active**. Also, if **OE:**

Customer Relationships is set to Yes, then the location must be the same as the ordering customer or a related customer.

SHIP_ADDRESS_ID	Conditionally Not Null	NUMBER
------------------------	------------------------	---------------

Enter the address ID for your location (represents your complete Ship To address for the order). This value is used to derive SO_LINES_INTERFACE.SHIP_TO_SITE_USE_ID. Refer to Table 6 – 12 on page 6 – 67 in the Using IDs section above for more details on this conditionally required for booking column.

Validation: Must represent a valid location for the SHIP_TO_CUSTOMER and exist in RA_SITE_USES where SITE_USE_CODE is **SHIP_TO**. Validates whether the address is valid and exists in RA_ADDRESSES; must be a valid address for the SHIP_TO_CUSTOMER. Also, if **OE: Customer Relationships** is set to Yes, then the location must be a valid site use for the Ship To customer and be the same as the ordering customer or a related customer.

SHIP_ADDRESS1 – 4	VARCHAR2(240)
--------------------------	----------------------

Enter the street address (4 lines) for your Ship To address. Refer to Table 6 – 12 on page 6 – 67 in the Using IDs section above for more details on this conditionally required for booking column.

SHIP_CITY	VARCHAR2(50)
------------------	---------------------

Enter the city for your Ship To address. Refer to Table 6 – 12 on page 6 – 67 in the Using IDs section above for more details on this conditionally required for booking column.

SHIP_STATE	VARCHAR2(50)
-------------------	---------------------

Enter the state or province for your Ship To address. Refer to Table 6 – 12 on page 6 – 67 in the Using IDs section above for more details on this conditionally required for booking column.

SHIP_COUNTY**VARCHAR2(50)**

Enter the county for your Ship To address. Refer to Table 6 – 12 on page 6 – 67 in the Using IDs section above for more details on this conditionally required for booking column.

SHIP_COUNTRY**VARCHAR2(50)**

Enter the country for your Ship To address. Refer to Table 6 – 12 on page 6 – 67 in the Using IDs section above for more details on this conditionally required for booking column.

SHIP_POSTAL_CODE**VARCHAR2(30)**

Enter the post code for your Ship To address. Refer to Table 6 – 12 on page 6 – 67 in the Using IDs section above for more details on this conditionally required for booking column.

PRICE_LIST_NAME**VARCHAR2(30)**

Enter the name of the price list for the order. If you are using IDs, you must enter the PRICE_LIST_ID. If you are not using IDs, then you must enter the PRICE_LIST_NAME, and if your PRICE_LIST_NAME is not unique, you must also enter the PRICE_LIST_ID.

Validation: SO_PRICE_LISTS.NAME

Destination: inserts SO_PRICE_LISTS.PRICE_LIST_ID into
SO_HEADERS.PRICE_LIST_ID where
SO_HEADER_INTERFACE.PRICE_LIST_NAME =
SO_PRICE_LISTS.NAME

PRICE_LIST_ID**NUMBER**

Enter the price list ID which matches your price list. If you are using IDs, you must enter the PRICE_LIST_ID. If you are not using IDs, then you must enter the PRICE_LIST_NAME, and if your PRICE_LIST_NAME is not unique, you must also enter the PRICE_LIST_ID.

Validation: SO_PRICE_LISTS.PRICE_LIST_ID

Destination: SO_HEADERS.PRICE_LIST_ID

TERMS_NAME**VARCHAR2(15)**

Enter the name of the payment terms for the order. If you are using IDs, you must enter the TERMS_ID. If you are not using IDs, then you

must enter the TERMS_NAME, and if your TERMS_NAME is not unique, you must also enter the TERMS_ID.

Validation: RA_TERMS.NAME

Destination: inserts RA_TERMS.TERM_ID into
SO_HEADERS.TERM_ID where
SO_HEADERS_INTERFACE.TERM_NAME =
RA_TERMS.NAME

TERMS_ID **NUMBER**

Enter the payment terms ID which matches your terms name. If you are using IDs, you must enter the TERMS_ID. If you are not using IDs, then you must enter the TERMS_NAME, and if your TERMS_NAME is not unique, you must also enter the TERMS_ID.

Validation: RA_TERMS.TERM_ID

Destination: SO_HEADERS.TERM_ID

CYCLE_ID **NUMBER**

The value for this column is derived from the order type.

Validation: None

Destination: SO_HEADERS.CYCLE_ID

ENTERED_
STATE_NAME Conditionally Not Null **VARCHAR2(30)**

Enter the state of the order when it is imported as **Entered**, **Partial**, or **Booked**. If you are using IDs, you should enter the ENTERED_STATE_ID. If you are not using IDs, then you should enter the ENTERED_STATE_NAME, and if your ENTERED_STATE_NAME is not unique, you should also enter the ENTERED_STATE_ID. If you do not supply a value for either the ENTERED_STATE_NAME or ENTERED_STATE_ID, Oracle Order Entry defaults the Entered State for the order to Entered.

Validation: **Entered, Partial, or Booked**

Destination: **Entered, Partial, and Booked** update
SO_HEADERS.S1 with **15, 5 OR 1**, respectively.

ENTERED_STATE_ID	Conditionally Not Null	NUMBER
-------------------------	------------------------	---------------

Enter the ID of the entered state of the order after OrderImport. If you are using IDs, you should enter the ENTERED_STATE_ID. If you are not using IDs, then you should enter the ENTERED_STATE_NAME, and if your ENTERED_STATE_NAME is not unique, you should also enter the ENTERED_STATE_ID. If you do not supply a value for either the ENTERED_STATE_NAME or ENTERED_STATE_ID, Oracle Order Entry defaults the Entered State for the order to Entered.

Validation: SO_RESULTS.RESULT_ID

ENTERED_STATE_DATE	DATE
---------------------------	-------------

Enter the date you wish to accompany the ENTERED_STATE_NAME. For example, if you enter **Booked** as the entered state name, this is the date it was booked.

Validation: Standard date validation

Destination: SO_HEADERS.S1_DATE.

COMPLETE_FLAG	VARCHAR2(1)
----------------------	--------------------

Enter **Y** to indicate if this order is already closed or should not be processed, in which case the order and order lines (but no schedule details) will be imported as a complete order. Complete orders are considered for historical purposes only and can have any entered state. Enter **N** or leave this column null to indicate that this is an open order.

AGREEMENT_NAME	VARCHAR2(30)
-----------------------	---------------------

Enter the name of the agreement for the order. This is only required if the order type is one which requires an agreement; otherwise it is an optional column. If you are using IDs, you can enter the AGREEMENT_ID. If you are not using IDs, then you can enter the AGREEMENT_NAME, and if your AGREEMENT_NAME is not unique, you must also enter the AGREEMENT_ID.

Validation: SO_AGREEMENT.NAME

Destination: inserts SO_AGREEMENTS.AGREEMENT_ID into SO_HEADERS.AGREEMENT_ID where SO_HEADERS_INTERFACE.AGREEMENT_NAME = SO_AGREEMENTS.NAME

AGREEMENT_ID	NUMBER
---------------------	---------------

Enter the agreement ID which matches your AGREEMENT_NAME. If you are using IDs, you can must the AGREEMENT_ID. If you are not using IDs, then you must enter the AGREEMENT_NAME, and if your AGREEMENT_NAME is not unique, you must also enter the AGREEMENT_ID.

Validation: SO_AGREEMENTS.AGREEMENT_ID

Destination: SO_HEADERS.AGREEMENT_ID

DATE_ REQUESTED_CURRENT	DATE
------------------------------------	-------------

Enter the date the customer requested receipt of the order.

Validation: Must be equal to or greater than
SO_HEADERS_INTERFACE.DATE_ORDERED

Destination: SO_HEADERS.DATE_REQUESTED_CURRENT

ORDERED_BY_ CONTACT_FIRST_NAME	VARCHAR2(30)
---	---------------------

Enter the first name of the contact who placed the order. Refer to Table 6 – 12 on page 6 – 67 in the Using IDs section above for more details on this conditionally required for booking column.

Validation: RA_CONTACTS.FIRST_NAME

ORDERED_BY_ CONTACT_LAST_NAME	VARCHAR2(50)
--	---------------------

Enter the last name of the contact who placed the order. Refer to Table 6 – 12 on page 6 – 67 in the Using IDs section above for more details on this conditionally required for booking column.

Validation: RA_CONTACTS.LAST_NAME. Both first and last names must be in RA_CONTACTS; the combination must match the ORDERED_BY_CONTACT_ID (if one is entered)

Destination: SO_HEADERS.ORDERED_BY_CONTACT_ID

**ORDERED_BY_
CONTACT_ID**

NUMBER

Enter the contact ID that matches your order contact. Refer to Table 6 – 12 on page 6 – 67 in the Using IDs section above for more details on this conditionally required for booking column.

Validation: RA_CONTACTS.CONTACT_ID.

Destination: SO_HEADERS.ORDERED_BY_CONTACT_ID

**SALES_
CHANNEL_CODE**

VARCHAR2(30)

Enter the name of the sales channel for the order.

Validation: SO_LOOKUPS.LOOKUP_CODE where
SO_LOOKUPS.LOOKUP_TYPE =
'SALES_CHANNEL'

Destination: SO_HEADERS.SALES_CHANNEL_CODE

PURCHASE_ORDER_NUM

VARCHAR2(50)

Enter the purchase order number for the order. This is only required if the order type is one which requires a purchase order, otherwise it is an optional column.

Validation: None

Destination: SO_HEADERS.PURCHASE_ORDER_NUM

**INVOICE_TO_
CONTACT_FIRST_NAME**

VARCHAR2(30)

Enter the first name of the contact at the invoice address. Refer to Table 6 – 12 on page 6 – 67 in the Using IDs section above for more details on this conditionally required for booking column.

Validation: RA_CONTACTS.FIRST_NAME

**INVOICE_TO_
CONTACT_LAST_NAME**

VARCHAR2(50)

Enter the last name of the contact at the invoice address. Refer to Table 6 – 12 on page 6 – 67 in the Using IDs section above for more details on this conditionally required for booking column.

Validation: RA_CONTACTS.LAST_NAME. Both first and last names must be in RA_CONTACTS; the combination must match the INVOICE_TO_CONTACT_ID (if one is entered)

INVOICE_TO_CONTACT_ID	NUMBER
------------------------------	---------------

Enter the contact ID for your Invoice To contact. Refer to Table 6 – 12 on page 6 – 67 in the Using IDs section above for more details on this conditionally required for booking column.

The combination of contact first and last name is validated.

Validation: RA_CONTACTS.CONTACT_ID

Destination: SO_HEADERS.INVOICE_TO_CONTACT_ID

SHIP_TO_CONTACT_FIRST_NAME	VARCHAR2(30)
-----------------------------------	---------------------

Enter the first name of the contact at the Ship To address. Refer to Table 6 – 12 on page 6 – 67 in the Using IDs section above for more details on this conditionally required for booking column.

SHIP_TO_CONTACT_LAST_NAME	VARCHAR2(50)
----------------------------------	---------------------

Enter the last name of the contact at the Ship To address. Refer to Table 6 – 12 on page 6 – 67 in the Using IDs section above for more details on this conditionally required for booking column.

SHIP_TO_CONTACT_ID	NUMBER
---------------------------	---------------

Enter the contact ID for your Ship To contact. Refer to Table 6 – 12 on page 6 – 67 in the Using IDs section above for more details on this conditionally required for booking column.

SHIPMENT_PRIORITY_CODE	VARCHAR2(30)
-------------------------------	---------------------

Enter the shipment priority for the order or use defaulting. The same defaulting used on the Enter Orders form is applied. Your standard value rule sets specify the source and priority of defaults.

Validation: SO_LOOKUPS_LOOKUP_CODE where
SO_LOOKUPS.LOOKUP_TYPE =
'SHIPMENT_PRIORITY'

Destination: SO_HEADERS.SHIPMENT_PRIORITY_CODE

SHIP_METHOD_CODE **VARCHAR2(30)**

Enter the freight carrier for the order or use defaulting. The same defaulting used on the Enter Orders form is applied. Your standard value rule sets specify the source and priority of defaults.

Validation: ORG_FREIGHT.FREIGHT_CODE

Destination: SO_HEADERS.SHIP_METHOD_CODE

FREIGHT_TERMS_CODE **VARCHAR2(30)**

Enter the freight terms for the order or use defaulting. The same defaulting used on the Enter Orders form is applied. Your standard value rule sets specify the source and priority of defaults.

Validation: SO_LOOKUPS.LOOKUP_CODE where
SO_LOOKUPS.LOOKUP_TYPE =
'FREIGHT_TERMS'

Destination: SO_HEADERS.FREIGHT_TERMS_CODE

FOB_CODE **VARCHAR2(30)**

Enter the FOB for the order or use defaulting. The same defaulting used on the Enter Orders form is applied. Your standard value rule sets specify the source and priority of defaults.

Validation: AR_LOOKUPS_LOOKUP_CODE where
AR_LOOKUPS.LOOKUP_TYPE = 'FOB'

Destination: SO_HEADERS.FOB_CODE

SHIPPING_INSTRUCTIONS **VARCHAR2(240)**

Enter any shipping instructions you wish to accompany the order.

Validation: None

Destination: SO_HEADERS.SHIPPING_INSTRUCTIONS

PACKING_INSTRUCTIONS **VARCHAR2(240)**

Enter any packing instructions you wish to accompany the order.

Validation: None

Destination: SO_HEADER.PACKING_INSTRUCTIONS

DATE_SHIPPED**DATE**

Enter the date the order was shipped, if you are importing the order in the **Complete** state. If this column is populated for open orders, this information is ignored.

Validation: Must be equal to or greater than
SO_HEADERS_INTERFACE.DATE_ORDERED

Destination: SO_HEADERS.DATE_SHIPPED

PAYMENT_TYPE_CODE**VARCHAR2(30)**

Enter the method of payment for the order.

Validation: AR_LOOKUPS.LOOKUP_CODE where
AR_LOOKUPS.LOOKUP_TYPE =
'CASH_RECEIPT_TYPE'

Destination: SO_HEADERS.PAYMENT_CODE

PAYMENT_AMOUNT**NUMBER**

Enter the payment amount for the order.

Validation: None

Destination: SO_HEADERS.PAYMENT_AMOUNT

CHECK_NUMBER**VARCHAR2(50)**

Enter the check number for the order.

Validation: PAYMENT_TYPE_CODE must be CHECK.

Destination: SO_HEADERS.CHECK_NUMBER

CREDIT_CARD_CODE**VARCHAR2(30)**

Enter the credit card code for the type of credit card the customer is using for payment.

Validation: PAYMENT_TYPE_CODE must be CREDIT CARD
and SO_LOOKUPS.LOOKUP_CODE where
SO_LOOKUPS.LOOKUP_TYPE =
'CREDIT_CARD'

Destination: SO_HEADERS.CREDIT_CARD

CREDIT_CARD_
HOLDER_NAME **VARCHAR2(50)**

Enter the name of the credit card holder.

Validation: PAYMENT_TYPE_CODE must be CREDIT CARD.

Destination: SO_HEADERS.CARDHOLDER_NAME

CREDIT_CARD_
NUMBER **VARCHAR2(50)**

Enter the credit card number.

Validation: PAYMENT_TYPE_CODE must be CREDIT CARD.

Destination: SO_HEADERS.CREDIT_CARD_NUM

CREDIT_CARD_
EXPIRATION_DATE **DATE**

Enter the expiration date of the credit card used for payment.

Validation: PAYMENT_TYPE_CODE must be CREDIT CARD
and the date must be equal to or greater than
SO_HEADERS_INTERFACE.DATE_ORDERED

Destination: SO_HEADERS.CREDIT_CARD_EXPIRATION_
DATE

CREDIT_CARD_
APPROVAL_CODE **VARCHAR2(50)**

Enter the credit card approval number.

Validation: PAYMENT_TYPE_CODE must be CREDIT CARD.

Destination: SO_HEADERS.CREDIT_CARD_APPROVAL_
CODE

ACCOUNTING_RULE **VARCHAR2(30)**

Enter the name of the accounting rule for the order, or use defaulting.
If you want to enter an accounting rule, and you are using IDs, you can
enter the ACCOUNTING_RULE_ID. If you are not using IDs, then
you must enter the ACCOUNTING_RULE name, and if your
ACCOUNTING_RULE is not unique, you must also enter the
ACCOUNTING_RULE_ID.

Validation: RA_RULES.NAME where RA_RULES.TYPE = 'A'

Destination: inserts RA_RULES.RULE_ID into
SO_HEADERS.ACCOUNTING_RULE where
RA_RULES.NAME =
SO_HEADERS.ACCOUNTING_RULE and
RA_RULES.TYPE = 'A'

<u>ACCOUNTING_RULE_ID</u>	<u>NUMBER</u>
---------------------------	---------------

Enter the rule ID that matches your accounting rule. If you want to enter an accounting rule, and you are using IDs, you can enter the ACCOUNTING_RULE_ID. If you are not using IDs, then you must enter the ACCOUNTING_RULE name, and if your ACCOUNTING_RULE is not unique, you must also enter the ACCOUNTING_RULE_ID.

Validation: RA_RULES.RULE_ID where
RA_RULES.TYPE = 'A'

Destination: SO_HEADERS.ACCOUNTING_RULE_ID

<u>INVOICING_RULE</u>	<u>VARCHAR2(30)</u>
-----------------------	---------------------

Enter the name of the invoicing rule for the order, or use defaulting. If you want to enter an invoicing rule, and you are using IDs, you can enter the INVOICING_RULE_ID. If you are not using IDs, then you must enter the INVOICING_RULE name, and if your INVOICING_RULE is not unique, you must also enter the INVOICING_RULE_ID.

Validation: RA_RULES.NAME where RA_RULES.TYPE = 'I'

Destination: inserts RA_RULES.RULE_ID into
SO_HEADERS.INVOICING_RULE where
RA_RULES.NAME =
SO_HEADERS.INVOICING_RULE and
RA_RULES.TYPE = 'I'

<u>INVOICING_RULE_ID</u>	<u>NUMBER</u>
--------------------------	---------------

Enter the rule ID that matches your invoicing rule. If you want to enter an invoicing rule, and you are using IDs, you can enter the INVOICING_RULE_ID. If you are not using IDs, then you must enter the INVOICING_RULE name, and if your INVOICING_RULE is not unique, you must also enter the INVOICING_RULE_ID.

Validation: RA_RULES.RULE_ID where
RA_RULES.TYPE = 'I'

Destination: SO_HEADERS.INVOICING_RULE_ID

CONTEXT	VARCHAR2(30)
----------------	---------------------

Enter the context for you descriptive flexfield, if you have enabled a context sensitive descriptive flexfield in SO_HEADERS.

Validation: None

Destination: SO_HEADERS.CONTEXT

ATTRIBUTE1 – 15	VARCHAR2(150)
------------------------	----------------------

Enter any information in these 15 columns that you wish to have imported into your descriptive flexfield columns in SO_HEADERS.

Validation: None

Destination: SO_HEADERS.ATTRIBUTE1 – 15

INTERFACE_STATUS	VARCHAR2(1000)
-------------------------	-----------------------

This column is used internally to store information about any invalid data.

ERROR_FLAG	VARCHAR2(1)
-------------------	--------------------

This column is used internally; **Y** means an error occurred when importing this header.

SCHEDULE_STATUS_CODE	VARCHAR2(30)
-----------------------------	---------------------

Enter the schedule status for the order; valid values include **DEMANDED** and **RESERVED**.

OPERATION_CODE	VARCHAR2(30)
-----------------------	---------------------

Enter **UPDATE** or **DELETE** for existing orders, or enter **INSERT** for new imported orders. You can also leave this column null for new imported orders.

SALESREP_NUMBER	NUMBER
------------------------	---------------

Enter the salesrep number if the salesrep name does not uniquely identify the salesperson. Refer to Table 6 – 12 on page 6 – 67 in the Using IDs section above for more details on this conditionally required for booking column.

TAX_EXEMPT_FLAG **VARCHAR2(1)**

Enter **E**, **S**, or **R** or accept the default. This stands for **Exempt**, **Standard** or **Required**, respectively.

Validation: If entered, must equal **E**, **S**, or **R**
AR_LOOKUPS.LOOKUP_CODE where
AR_LOOKUPS.LOOKUP_TYPE =
'TAX_CONTROL_FLAG'

Destination: SO_HEADERS.TAX_EXEMPT_FLAG

TAX_EXEMPT_NUM **VARCHAR2(30)**

If you enter **E** in the TAX_EXEMPT_FLAG field, you can enter an existing certificate number for the ship to customer or enter a new, unapproved exemption certificate number.

Validation: Must be NULL if you enter Standard or Required
in the Tax Exempt Flag

Destination: SO_HEADERS.TAX_EXEMPT_NUM

TAX_EXEMPT_REASON_CODE **VARCHAR2(30)**

If you enter **E** in the TAX_EXEMPT_FLAG field, you must enter a valid tax exempt reason code.

Validation: AR_LOOKUPS.LOOKUP_CODE where
AR_LOOKUPS.LOOKUP_TYPE = 'TAX_REASON'

Destination: SO_HEADERS.TAX_EXEMPT_REASON_CODE

APPLY_STANDARD_NOTES **VARCHAR2(1)**

Enter **Y** to automatically apply standard notes.

ORG_ID **NUMBER**

Reserved for future enhancement.

SO_LINES_INTERFACE

The following graphic summarizes the SO_LINES_INTERFACE table. Database columns with a check under Conditionally Required for Booking are optional if you are importing an order with an Entry Status

of **Entered**. If you are importing an order with an Entry Status of **Booked**, you must enter the ID column if you are using IDs, or the name column if you are not using IDs. Refer to the Using IDs section on page 6 – 66, and the Required Fields for Booked Orders section on page 6 – 71 for more details.

Database columns not included in this table are for internal use only. Each column is described in detail below starting on page 6 – 103.

SO_LINES_INTERFACE Column Name	Type	Required (C indicates Condi- tionally Required)	Condi- tionally Required for Booking	Derived	Optional
CREATION_DATE	Date	✓			
CREATED_BY	Number	✓			
LAST_UPDATE_DATE	Date	✓			
LAST_UPDATED_BY	Number	✓			
LAST_UPDATE_LOGIN	Number				✓
PROGRAM_ APPLICATION_ID	Number				✓
PROGRAM_ID	Number				✓
PROGRAM_UPDATE_DATE	Date				✓
REQUEST_ID	Number				✓
ORIGINAL_SYSTEM_ REFERENCE	Var- char2(50)	✓			
ORIGINAL_SYSTEM_ LINE_REFERENCE	Var- char2(50)	✓			
LINE_NUMBER	Number	✓			
LINE_TYPE	Var- char2(30)	✓			
UNIT_CODE	Varchar2(3)	✓			
ORDERED_QUANTITY	Number	✓			
DATE_ REQUESTED_CURRENT	Date		✓		
LIST_PRICE	Number	C			

Table 6 – 14 SO_LINES_INTERFACE Table (Page 1 of 4)

SO_LINES_INTERFACE Column Name	Type	Required (C indicates Condi- tionally Required)	Condi- tionally Required for Booking	Derived	Optional
SELLING_PRICE	Number	C			
INVENTORY_ITEM_ SEGMENT1 - 20	Var- char2(40)	C			
INVENTORY_ITEM_ID	Number	C			
SHIPPED_QUANTITY	Number				✓
SCHEDULED_ SHIPMENT_DATE	Date				✓
LINK_TO_LINE_REF	Var- char2(50)				✓
PARENT_LINE_REF	Var- char2(50)				✓
SHIP_SET_NUMBER	Number				✓
SHIP_TO_SITE_USE_ID	Number			✓	
SHIP_TO_CUSTOMER	Var- char2(50)		✓		
SHIP_TO_CUSTOMER_ID	Number		✓		
SHIP_TO_CONTACT_ID	Number				✓
SHIPMENT_ PRIORITY_CODE	Var- char2(30)				✓
SHIP_METHOD_CODE	Var- char2(30)				✓
WAREHOUSE_ID	Number				✓
AGREEMENT_NAME	Var- char2(30)			✓	
AGREEMENT_ID	Number			✓	
ACCOUNTING_RULE	Var- char2(30)				✓
ACCOUNTING_RULE_ID	Number				✓

Table 6 – 14 SO_LINES_INTERFACE Table (Page 2 of 4)

SO_LINES_INTEREACE Column Name	Type	Required (C indicates Condi- tionally Required)	Condi- tionally Required for Booking	Derived	Optional
INVOICING_RULE	Var- char2(30)				✓
INVOICING_RULE_ID	Number				✓
ORDER_CATEGORY	Var- char2(30)				✓
ENTERED_STATE	Var- char2(30)			✓	
ENTERED_STATE_DATE	Date			✓	
COMPLETE_FLAG	Varchar2(1)			✓	
CONTEXT	Var- char2(30)				✓
ATTRIBUTE1 – 15	Var- char2(150)				✓
CALCULATE_PRICE	Varchar2(1)				✓
PRICING_ATTRIBUTE1 – 15	Var- char2(150)				✓
PRICING_CONTEXT	Var- char2(30)				✓
PRICING_METHOD_CODE	Var- char2(30)				✓
ITEM_TYPE_CODE	Var- char2(30)				✓
OPTION_FLAG	Varchar2(1)				✓
ORDER_SOURCE_ID	Number	✓			
SHIP_ADDRESS1 – 4	Var- char2(240)		✓		
SHIP_ADDRESS_ID	Number		✓		
SHIP_CITY	Var- char2(50)		✓		

Table 6 – 14 SO_LINES_INTERFACE Table (Page 3 of 4)

SO_LINES_INTERFACE Column Name	Type	Required (C indicates Condi- tionally Required)	Condi- tionally Required for Booking	Derived	Optional
SHIP_COUNTRY	Var- char2(50)		✓		
SHIP_COUNTY	Var- char2(50)		✓		
SHIP_POSTAL_CODE	Var- char2(30)		✓		
SHIP_STATE	Var- char2(50)		✓		
SHIP_TO_ CONTACT_FIRST_NAME	Var- char2(30)				✓
SHIP_TO_ CONTACT_LAST_NAME	Var- char2(50)				✓
OPERATION_CODE	Var- char2(30)				✓
COMMITMENT_ID	Number				✓
TAX_CODE	Var- char2(50)	C			

Table 6 – 14 SO_LINES_INTERFACE Table (Page 4 of 4)

CREATION_DATE Not Null **DATE**

Enter the date that you originally entered the data into your feeder system.

Validation: Standard Date Validation

Destination: SO_LINES.CREATION_DATE

CREATED_BY Not Null **NUMBER**

Enter an identification number that you can use to identify the user who created the record.

Validation: None

Destination: SO_LINES.CREATED_BY

LAST_UPDATE_DATE	Not Null	DATE
-------------------------	----------	-------------

Enter the current date that you are entering data into your feeder system.

Validation: Standard Date Validation

Destination: SO_LINES.LAST_UPDATE_DATE

LAST_UPDATED_BY	Not Null	NUMBER
------------------------	----------	---------------

Enter an identification number that you can use to identify the user who created or who most recently modified the record.

Validation: None

Destination: SO_LINES.LAST_UPDATE_BY

LAST_UPDATE_LOGIN		NUMBER
--------------------------	--	---------------

Enter an identification number that you can use to identify the feeder system that is supplying Oracle Order Entry with this data.

Validation: None

Destination: SO_LINES.LAST_UPDATE_LOGIN

PROGRAM_APPLICATION_ID		NUMBER(15)
-------------------------------	--	-------------------

This column is for internal use only.

PROGRAM_ID		NUMBER(15)
-------------------	--	-------------------

This column is for internal use only.

PROGRAM_UPDATE_DATE		DATE
----------------------------	--	-------------

This column is for internal use only.

REQUEST_ID		NUMBER
-------------------	--	---------------

This column is for internal use only. It is populated with the concurrent manager request ID for the run of the interface each time you submit the OrderImport program. In order for records to be processed, their request ID must be null or equal to the request ID being processed.

ORIGINAL_		
SYSTEM_REFERENCE	Not Null	VARCHAR2(50)

Enter the order number or ID from your feeder system.

Validation: Matches
SO_HEADERS.ORIGINAL_SYSTEM_REFERENCE

Destination: SO_HEADERS.ORIGINAL_SYSTEM_REFERENCE

ORIGINAL_SYSTEM_		
LINE_REFERENCE	Not Null	VARCHAR2(50)

Enter the line ID from your feeder system.

Validation: No repeated line IDs allowed on an order.

Destination: SO_LINES.ORIGINAL_SYSTEM_LINE_
REFERENCE

LINE_NUMBER	Not Null	NUMBER
--------------------	----------	---------------

Enter the line number of the order line. The line number can be from the original system.

Validation: Must be greater than zero, an integer, and unique within the order.

Destination: SO_LINES.LINE_NUMBER

LINE_TYPE	Not Null	VARCHAR2(30)
------------------	----------	---------------------

Enter **REGULAR**. Shipment schedules are not supported.

Validation: Must be **REGULAR**

Destination: SO_LINES.LINE_TYPE_CODE

UNIT_CODE	Not Null	VARCHAR2(3)
------------------	----------	--------------------

Enter the unit of measure code for the order line.

Validation: MTL_ITEM_UOMS_VIEW.UOM_CODE

Destination: SO_LINES.UNIT_CODE

ORDERED_QUANTITY	Not Null	NUMBER
-------------------------	----------	---------------

Enter the quantity ordered of this line item.

Validation: Must be greater than zero and an integer

Destination: SO_LINES.ORDERED_QUANTITY

DATE_
REQUESTED_CURRENT **DATE**

Enter the date the customer requested receipt of the order or use defaulting.

Validation: Must be equal to or greater than
SO_HEADERS.DATE_ORDERED

Destination: SO_LINES.DATE_REQUEST_CURRENT

LIST_PRICE (Conditionally Required) **NUMBER**

Enter the list price per unit for the order line or use automatic pricing. This column is required if you enter No in the SO_LINES_INTERFACE.CALCULATE_PRICE column.

Validation: SO_PRICE_LIST_LINES

Destination: SO_LINES.LIST_PRICE

SELLING_PRICE (Conditionally Required) **NUMBER**

Enter the selling price per unit for the order line or use automatic pricing. This column is required if you enter No in the SO_LINES_INTERFACE.CALCULATE_PRICE column.

Validation: Must be less than or equal to the LIST_PRICE

Destination: SO_LINES.SELLING_PRICE

LIST_PERCENT **NUMBER**

This column is for internal use only.

SELLING_PERCENT **NUMBER**

This column is for internal use only.

INVENTORY_ITEM_
SEGMENT1 – 20 Conditionally Required **VARCHAR2(40)**

Enter the segments of the inventory item you wish to order on the order line. As long as you enter the segments in the correct sequence, OrderImport knows which segment columns to place the values in. For example, if you use three segments, enter values in segments 1 – 3.

You must enter either the INVENTORY_ITEM_SEGMENT columns or INVENTORY_ITEM_ID, depending on whether you are using IDs.

Validation: Valid code combination in MTL_SYSTEM_ITEMS

Destination: insert
MTL_SYSTEM_ITEMS.INVENTORY_ITEM_ID
into SO_LINES.INVENTORY_ITEM_ID where
SO_LINES_INTERFACE.INVENTORY_
ITEM_SEGMENT1 - N =
MTL_SYSTEM_ITEMS.SEGMENTS1 - N

INVENTORY_ITEM_ID Conditionally Required **NUMBER**

Enter the inventory item code combination ID which matches your combination of inventory item segments. You must enter either the INVENTORY_ITEM_SEGMENT columns or INVENTORY_ITEM_ID, depending on whether you are using IDs.

Validation: MTL_SYSTEM_ITEMS.INVENTORY_ITEM_ID

Destination: SO_LINES.INVENTORY_ITEM_ID

SHIPPED_QUANTITY **NUMBER**

Enter the shipped quantity, if you are importing this order as complete. If you enter a shipped quantity on an open order it is ignored.

Validation: None

Destination: SO_LINES.SHIPPED_QUANTITY

SCHEDULED_SHIPMENT_DATE **DATE**

Enter the scheduled shipment date for the order line or use defaulting.

Validation: Must be greater than or equal to
SO_HEADERS_INTERFACE.DATE_ORDERED

Destination: SO_LINES.SCHEDULED_SHIPMENT_DATE

LINK_TO_LINE_REF **VARCHAR2(50)**

Enter the Original System Line Reference of the line for the item that is immediately above the item in this line in the bill of material (BOM) structure. This only needs to be populated for option items, option classes and kits. (This does not need to be entered for the top level item.)

PARENT_LINE_REF	VARCHAR2(50)
------------------------	---------------------

Enter the Original System Line Reference of the line for the item that is the top level item of the BOM to which the item in this line belongs. This only needs to be populated for option items, option classes and optional standard items. (This does not need to be entered for the top level item.)

SHIPMENT_ SCHEDULE_LINE_REF	VARCHAR2(50)
--	---------------------

This column is for internal use only.

SHIP_SET_NUMBER	NUMBER
------------------------	---------------

Use this column if you want some lines to ship together (they belong to the same ship set). If two or more lines have the same ship set number, they belong to one ship set. This only needs to be populated for lines which belong to a ship set.

Validation: Must be greater than zero

Destination: SO_LINES.SHIP_SET_NUMBER

SHIP_TO_SITE_USE_ID	NUMBER
----------------------------	---------------

Leave this column null, it is derived from the SHIP_ADDRESS_ID column.

Destination: SO_LINES.SHIP_TO_SITE_USE_ID

SHIP_TO_CUSTOMER	NUMBER
-------------------------	---------------

Enter the name of the customer to receive the shipment, if it is different from the order. Refer to Table 6 – 12 on page 6 – 67 in the Using IDs section above for more details on this conditionally required for booking column.

Validation: Validates whether the SHIP_TO_CUSTOMER exists in RA_CUSTOMERS and RA_CUSTOMERS. STATUS is **Active**. (If the customer name is not unique, you must enter an ID in the SHIP_TO_CUSTOMER_ID column.) Also, if **OE: Customer Relationships** is set to Yes, then the location must be the same as the ordering customer or a related customer.

Destination: None

SHIP_TO_ CUSTOMER_ID	Conditionally Not Null	NUMBER
---------------------------------	------------------------	---------------

Enter the customer ID of your Ship To Customer, if it is different from the order. If you are using IDs, you must enter the SHIP_TO_CUSTOMER_ID. If you are not using IDs, then you must enter the SHIP_TO_CUSTOMER. Refer to Table 6 – 12 on page 6 – 67 in the Using IDs section above for more details on this conditionally required for booking column.

Validation: Validates whether the SHIP_TO_CUSTOMER_ID exists in RA_CUSTOMERS and RA_CUSTOMERS.STATUS is **Active**. Also, if OE: **Customer Relationships** is set to Yes, then the location must be the same as the ordering customer or a related customer.

SHIP_TO_CONTACT_ID	NUMBER
---------------------------	---------------

Enter the contact ID for your Ship To contact.

Validation: RA_CONTACTS.CONTACT_ID

SHIPMENT_ PRIORITY_CODE	VARCHAR2(30)
------------------------------------	---------------------

Enter the shipment priority for the line or use defaulting. The same defaulting used on the Enter Orders form is applied. Your standard value rule sets specify the source and priority of defaults.

Validation: SO_LOOKUPS_LOOKUP_CODE where
SO_LOOKUPS.LOOKUP_TYPE =
'SHIPMENT_PRIORITY'

Destination: SO_LINES.SHIPMENT_PRIORITY_CODE

SHIP_METHOD_CODE	VARCHAR2(30)
-------------------------	---------------------

Enter the freight carrier for the line or use defaulting. The same defaulting used on the Enter Orders form is applied. Your standard value rule sets specify the source and priority of defaults.

Validation: ORG_FREIGHT.FREIGHT_CODE

Destination: SO_LINES.SHIP_METHOD_CODE

PRICE_LIST_ID**NUMBER**

This column is for internal use only.

TERMS_ID**NUMBER**

This column is for internal use only.

WAREHOUSE_ID**NUMBER**

Enter the name of the warehouse from which you will ship this item or use defaulting. The same defaulting used on the Enter Orders form is applied. Your standard value rule sets specify the source and priority of defaults.

Validation: The warehouse specified must have the item attribute **Customer Orders Enabled** set to **Yes** for the order line item.

ORG_ORGANIZATION_
DEFINITIONS.ORGANIZATION_ID

Destination: SO_LINES.WAREHOUSE_ID

SERVICED_SEGMENT1 – 20**VARCHAR2(40)**

These columns are for internal use only.

SERVICED_ITEM_ID**NUMBER**

This column is for internal use only.

SERVICED_LIST_PRICE**NUMBER**

This column is for internal use only.

SERVICE_START_DATE_ACTIVE**DATE**

This column is for internal use only.

SERVICE_END_DATE_ACTIVE**DATE**

This column is for internal use only.

CUSTOMER_PRODUCT_ID**NUMBER**

This column is for internal use only.

CP_SERVICE_ID	NUMBER
----------------------	---------------

This column is for internal use only.

TRANSACTION_TYPE_CODE	VARCHAR2(30)
------------------------------	---------------------

This column is for internal use only.

SERVICE_ DURATION_QUANTITY	NUMBER
---------------------------------------	---------------

This column is for internal use only.

TRANSACTION_ REASON_CODE	VARCHAR2(30)
-------------------------------------	---------------------

This column is for internal use only.

TRANSACTION_COMMENTS	VARCHAR2(240)
-----------------------------	----------------------

This column is for internal use only.

SERVICE_ MASS_TXN_TEMP_ID	NUMBER
--------------------------------------	---------------

This column is for internal use only.

AGREEMENT_NAME	VARCHAR2(30)
-----------------------	---------------------

Leave this column null, it is derived from
SO_HEADERS_INTERFACE.AGREEMENT_NAME.

AGREEMENT_ID	NUMBER
---------------------	---------------

Leave this column null, it is derived from
SO_HEADERS_INTERFACE.AGREEMENT_ID.

ACCOUNTING_RULE	VARCHAR2(30)
------------------------	---------------------

Enter the name of the accounting rule only if this is a complete order.
Otherwise leave this column null and use defaulting with your
standard value rule sets to assure that the order line has an accounting
rule.

Validation: RA_RULES.NAME where RA_RULES.TYPE = 'A'

Destination: inserts RA_RULES.RULE_ID into
SO_LINES.ACCOUNTING_RULE_ID where
RA_RULES.NAME =
SO_LINES.ACCOUNTING_RULE_ID and
RA_RULES.TYPE = 'A'

ACCOUNTING_RULE_ID	NUMBER
---------------------------	---------------

Enter the accounting rule ID only if this is a complete order. Otherwise leave this column null and use defaulting with your standard value rule sets to assure that the order line has an accounting rule.

Validation: RA_RULES.RULE_ID where
RA_RULES.TYPE = 'A'

Destination: SO_LINES.ACCOUNTING_RULE_ID

INVOICING_RULE	VARCHAR2(30)
-----------------------	---------------------

Enter the name of the invoicing rule only if this is a complete order. Otherwise leave this column null and use defaulting with your standard value rule sets to assure that the order line has an invoicing rule.

Validation: RA_RULES.NAME where RA_RULES.TYPE = 'I'

Destination: inserts RA_RULES.RULE_ID into
SO_LINES.INVOICING_RULE_ID where
RA_RULES.NAME =
SO_LINES.INVOICING_RULE_ID and
RA_RULES.TYPE = 'I'

INVOICING_RULE_ID	NUMBER
--------------------------	---------------

Enter the invoicing rule ID only if this is a complete order. Otherwise leave this column null and use defaulting with your standard value rule sets to assure that the order line has an invoicing rule.

Validation: RA_RULES.RULE_ID where
RA_RULES.TYPE = 'I'

Destination: SO_LINES.INVOICING_RULE_ID

SERVICED_UNIT_CODE	VARCHAR2(3)
---------------------------	--------------------

This column is for internal use only.

SERVICED_PRICING_ ATTRIBUTE1 – 15	VARCHAR2(150)
--	----------------------

These columns are for internal use only.

SERVICED_PRICING_ CONTEXT	VARCHAR2(30)
--	---------------------

This column is for internal use only.

ORDER_CATEGORY	VARCHAR2(30)
-----------------------	---------------------

Enter **R**, for Regular order line, in this column.

Validation: Must equal **R**.

ENTERED_STATE	VARCHAR2(30)
----------------------	---------------------

Leave this column null, it is derived from SO_HEADERS_INTERFACE.

ENTERED_STATE_DATE	DATE
---------------------------	-------------

Leave this column null, it is derived from SO_HEADERS_INTERFACE.

COMPLETE_FLAG	VARCHAR2(1)
----------------------	--------------------

Leave this column null, it is derived from SO_HEADERS_INTERFACE.

CONTEXT	VARCHAR2(30)
----------------	---------------------

Enter the context for your descriptive flexfield attributes if you have enabled a context sensitive descriptive flexfield on SO_LINES.

Validation: None

Destination: SO_LINES.CONTEXT

ATTRIBUTE1 – 15	VARCHAR2(150)
------------------------	----------------------

Enter any information you wish to pass to the descriptive flexfield columns in SO_LINES.

Validation: None

Destination: SO_LINES.ATTRIBUTE1 – 15

Validation: None

Destination SO_LINES.SHIP_SET_NUMBER

CALCULATE_PRICE**VARCHAR2(1)**

Enter **N** if the you are providing a list and selling price in the **LIST_PRICE** and **SELLING_PRICE** columns, or enter **Y** if you want Oracle Order Entry to automatically price the order line.



Attention: If you enter **N**, and the values you enter for the list and selling prices are not equal, then you must provide a price adjustment for the order line for the difference.

Validation: None

PRICING_ATTRIBUTE1 – 15**VARCHAR2(150)**

Enter any information you wish to pass to the pricing descriptive flexfield columns in **SO_LINES**.

Validation: None

Destination: **SO_LINES.PRICING_ATTRIBUTE1 – 15**

PRICING_CONTEXT**VARCHAR2(30)**

Enter the context for your pricing descriptive flexfield attributes if you have enabled a context sensitive pricing descriptive flexfield on **SO_LINES**.

Validation: None

Destination: **SO_LINES.PRICING_CONTEXT**

PRICING_METHOD_CODE**VARCHAR2(30)**

Enter **AMOUNT** in this column.

COMPONENT_CODE**VARCHAR2(240)**

This column is for internal use only.

COMPONENT_SEQUENCE_ID**NUMBER**

This column is for internal use only.

COMPONENT_SORT_CODE**VARCHAR2(240)**

This column is for internal use only.

ERROR_FLAG	VARCHAR2(1)
-------------------	--------------------

This column is used internally; **Y** means an error occurred when importing this header.

INTERFACE_STATUS	VARCHAR2(1000)
-------------------------	-----------------------

This column is for internal use. It contains error information if order import failed to import this order line.

ITEM_TYPE_CODE	VARCHAR2(30)
-----------------------	---------------------

Enter **CLASS**, **KIT**, **MODEL**, or **STANDARD** to indicate the type of item for the order line.

Validation: DECODE(MTL_SYSTEM_ITEMS.PICK_COMPONENTS_FLAG, 'Y',
 DECODE(MTL_SYSTEM_ITEMS.BOM_ITEM_TYPE, 1, 'MODEL', 2, 'CLASS', 4, 'KIT'),
 DECODE(MTL_SYSTEM_ITEMS.BOM_ITEM_TYPE, 1, 'MODEL', 2, 'CLASS', 4, 'STANDARD')) =
 SO_LINES_INTERFACE.ITEM_TYPE_CODE

LEVEL_CODE	NUMBER
-------------------	---------------

This column is for internal use only.

OPTION_FLAG	VARCHAR2(1)
--------------------	--------------------

Enter **Y** to indicate that the item on this line is an option item or option class. Enter **N** or null for base models or regular items.

ORDER_SOURCE_ID	Not Null	NUMBER
------------------------	----------	---------------

Enter the order import source ID (this should match the ORDER_SOURCE_ID of the order for this line). This column tells OrderImport whether to use IDs or the name columns and determines which records in the interface table to process.

Validation: SO_ORDER_SOURCES.ORDER_SOURCE_ID

Destination: None



Define OrderImport Sources
(Oracle Order Entry Reference Manual)

SHIP_ADDRESS1 – 4**VARCHAR2(240)**

Enter the street address (using up to 4 lines, one in each column) for your Ship To address for the order line, if it is different from the order.

SHIP_ADDRESS_ID**NUMBER**

Enter the address ID for your location (represents your complete Ship To address for the order line) if it is different from the order. This value is used to derive SO_LINES_INTERFACE.SHIP_TO_SITE_USE_ID.

Validation:

Must represent a location contained in RA_SITE_USES where SITE_USE_CODE = 'SHIP_TO'. Validates whether the address is valid and exists in RA_ADDRESSES; must be a valid address for SO_LINES_INTERFACE.SHIP_TO_CUSTOMER, if entered, otherwise it must be valid for SO_HEADERS_INTERFACE.SHIP_TO_CUSTOMER. Also, if **OE: Customer Relationships** is set to Yes, then the location must be a valid site use for the Ship To customer and be the same as the ordering customer or a related customer.

SHIP_CITY**VARCHAR2(50)**

Enter the city for your Ship To address for the order line, if it is different from the order.

SHIP_COUNTRY**VARCHAR2(50)**

Enter the country for your Ship To address for the order line, if it is different from the order.

SHIP_COUNTY**VARCHAR2(50)**

Enter the county for your Ship To address for the order line, if it is different from the order.

SHIP_POSTAL_CODE**VARCHAR2(30)**

Enter the post code for your Ship To address for the order line, if it is different from the order.

SHIP_STATE **VARCHAR2(50)**

Enter the state or province for your Ship To address for the order line, if it is different from the order.

SHIP_TO_
CONTACT_FIRST_NAME **VARCHAR2(30)**

Enter the first name of the contact at the Ship To address for the order line, if it is different from the order.

SHIP_TO_
CONTACT_LAST_NAME **VARCHAR2(50)**

Enter the last name of the contact at the Ship To address for the order line, if it is different from the order.

TERMS_NAME **VARCHAR2(15)**

This column is for internal use only.

GROUP_ID **NUMBER**

This column is for internal use only.

PERCENT_BASE_PRICE **NUMBER**

This column is for internal use only.

OPERATION_CODE **VARCHAR2(30)**

Enter **UPDATE** or **DELETE** for existing order lines, or enter **INSERT** for new imported order lines. You can also leave this column null for new imported order lines.

COMMITMENT_ID **NUMBER**

Enter the commitment ID for the order line.

Validation: RA_CUSTOMER_TRX.CUSTOMER_TRX_ID

Destination: SO_LINES.COMMITMENT_ID

TAX_CODE**VARCHAR2(50)**

If Value Added Tax is applicable, enter the tax code for the order line or accept the default. Tax code is required if the Tax Exempt Flag on the order is **R** or the Receivables Transaction Type associated with the order type requires tax calculation.

Validation: Must represent a valid tax code in Oracle Receivables
 AR_VAT_TAX V, AR_SYSTEM_PARAMETERS P
 WHERE V.SET_OF_BOOKS_ID =
 P.SET_OF_BOOKS_ID AND ((P.TAX_METHOD =
 'VAT' AND V.TAX_TYPE != 'LOCATION')
 OR (P.TAX_METHOD = 'SALES_TAX')) AND
 NVL(SO_LINES_INTERFACE.TAX_CODE,V.TAX_
 CODE) = V.TAX_CODE AND
 NVL(V.START_DATE,SYSDATE) AND
 NVL(V.END_DATE,SYSDATE)

Destination: SO_LINES.TAX_CODE may be NULL if the line does not require tax.

SO_LINE_DETAILS_INTERFACE

The following graphic summarizes the SO_LINE_DETAILS_INTERFACE table. Database columns with a check under Conditionally Required for Booking are optional if you are importing an order with an Entry Status of **Entered**. If you are importing an order with an Entry Status of **Booked**, you must enter the ID column if you are using IDs, or the name column if you are not using IDs. Refer to the Using IDs section on page 6 – 66, and the Required Fields for Booked Orders section on page 6 – 71 for more details.

Database columns not included in this table are for internal use only. Each column is described in detail below starting on page 6 – 120.

SO_LINE_DETAILS_INTERFACE Column Name	Type	Required (C indicates Condi- tionally Required)	Condi- tionally Required for Booking	Derived	Optional
CREATION_DATE	Date	✓			
CREATED_BY	Number	✓			

SO_LINE_DETAILS_ INTERFACE Column Name	Type	Required (C indicates Condi- tionally Required)	Condi- tionally Required for Booking	Derived	Optional
LAST_UPDATE_DATE	Date	✓			
LAST_UPDATED_BY	Number	✓			
LAST_UPDATE_LOGIN	Number				✓
PROGRAM_ APPLICATION_ID	Number				✓
PROGRAM_ID	Number				✓
PROGRAM_UPDATE_DATE	Date				✓
REQUEST_ID	Number				✓
ORDER_SOURCE_ID	Number	✓			
ORIGINAL_SYSTEM_ REFERENCE	Var- char2(50)	✓			
ORIGINAL_SYSTEM_ LINE_REFERENCE	Var- char2(50)	✓			
QUANTITY	Number	✓			
SCHEDULE_DATE	Date		✓		
LOT_NUMBER	Var- char2(30)		✓		
SUBINVENTORY	Var- char2(10)		✓		
CUSTOMER_ REQUESTED_LOT_FLAG	Varchar2(1)				✓
CONTEXT	Var- char2(30)				✓
ATTRIBUTE1 – 15	Var- char2(150)				✓
INVENTORY_ITEM_ID	Number			✓	

Table 6 – 15 SO_LINE_DETAILS_INTERFACE Table (Page 2 of 3)

SO_LINE_DETAILS_INTERFACE Column Name	Type	Required (C indicates Condi- tionally Required)	Condi- tionally Required for Booking	Derived	Optional
REVISION	Varchar2(3)		✓		
WAREHOUSE_ID	Number		✓		

Table 6 – 15 SO_LINE_DETAILS_INTERFACE Table (Page 3 of 3)

CREATION_DATE Not Null **DATE**

Enter the date that you originally entered the data into your feeder system.

Validation: Standard Date Validation

Destination: SO_LINES.CREATION_DATE

CREATED_BY Not Null **NUMBER**

Enter an identification number that you can use to identify the user who created the record.

Validation: None

Destination: SO_LINES.CREATED_BY

LAST_UPDATE_DATE Not Null **DATE**

Enter the current date that you are entering data into your feeder system.

Validation: Standard Date Validation

Destination: SO_LINES.LAST_UPDATE_DATE

LAST_UPDATED_BY Not Null **NUMBER**

Enter an identification number that you can use to identify the user who created or who most recently modified the record.

Validation: None

Destination: SO_LINES.LAST_UPDATE_BY

LAST_UPDATE_LOGIN	NUMBER
--------------------------	---------------

Enter an identification number that you can use to identify the feeder system that is supplying Oracle Order Entry with this data.

Validation: None

Destination: SO_LINES.LAST_UPDATE_LOGIN

PROGRAM_APPLICATION_ID	NUMBER
-------------------------------	---------------

This column is for internal use only.

PROGRAM_ID	NUMBER
-------------------	---------------

This column is for internal use only.

PROGRAM_UPDATE_DATE	DATE
----------------------------	-------------

This column is for internal use only.

REQUEST_ID	NUMBER
-------------------	---------------

This column is for internal use only. It is populated with the concurrent manager request ID for the run of the interface each time you submit the OrderImport program. In order for records to be processed, their request ID must be null or equal to the request ID being processed.

ORDER_SOURCE_ID	Not Null	NUMBER
------------------------	----------	---------------

Enter the order import source ID (this should match the ORDER_SOURCE_ID of the line for which this is a line detail). This column tells OrderImport whether to use IDs or the name columns and determines which records in the interface table to process.

Validation: SO_ORDER_SOURCES.ORDER_SOURCE_ID

Destination: None

ORIGINAL_SYSTEM_REFERENCE	Not Null	VARCHAR2(50)
----------------------------------	----------	---------------------

Enter the order number or ID from your feeder system.

Validation: Matches SO_HEADERS_INTERFACE.ORIGINAL_SYSTEM_REFERENCE

Destination: SO_LINE_DETAILS.ORIGINAL_SYSTEM_REFERENCE

ORIGINAL_SYSTEM_
LINE_REFERENCE Not Null **VARCHAR2(50)**

Enter the line ID from your feeder system.

Validation: Matches
SO_LINES_INTERFACE.ORIGINAL_SYSTEM_
LINE_REFERENCE

Destination: SO_LINE_DETAILS.ORIGINAL_SYSTEM_LINE_REFERENCE

QUANTITY Not Null **NUMBER**

Enter the quantity for the order line detail.

SCHEDULE_DATE **DATE**

Enter the schedule date for the order line detail.

LOT_NUMBER **VARCHAR2(30)**

Enter the lot number to use when releasing item from inventory if you entered **Reserved** in SO_HEADERS_INTERFACE.SCHEDULE_STATUS_CODE. Otherwise leave this column null.

SUBINVENTORY **VARCHAR2(10)**

Enter the subinventory to use when releasing item from inventory.

CUSTOMER_
REQUESTED_LOT_FLAG **VARCHAR2(1)**

Enter Y or N depending on whether the customer requested a specific subinventory, lot or revision.

CONTEXT **VARCHAR2(30)**

Enter the context for your descriptive flexfield attributes if you have enabled a context sensitive descriptive flexfield on SO_LINE_DETAILS.

Validation: None

Destination: SO_LINE_DETAILS.CONTEXT

ATTRIBUTE1 – 15 **VARCHAR2(150)**

Enter any information you wish to pass to the descriptive flexfield columns in SO_LINE_DETAILS.

Validation: None

Destination: SO_LINE_DETAILS.ATTRIBUTE1 – 15

COMPONENT_SEQUENCE_ID **NUMBER**

This column is for internal use only.

INVENTORY_ITEM_ID **NUMBER**

Leave this column null, it is derived from SO_LINES_INTERFACE.

REVISION **VARCHAR2(3)**

Enter the revision number of item being scheduled if you entered **Reserved** in SO_HEADERS_INTERFACE.SCHEDULE_STATUS_CODE. Otherwise leave this column null.

SCHEDULE_LEVEL_CODE **NUMBER**

This column is for internal use only.

WAREHOUSE_ID **NUMBER**

Enter the name of the warehouse from which you will ship this item or use defaulting. The same defaulting used on the Enter Orders form is applied. Your standard value rule sets specify the source and priority of defaults.

Validation: The warehouse specified must have the item attribute **Customer Orders Enabled** set to **Yes** for the order line item.
ORG_ORGANIZATION_DEFINITIONS.
ORGANIZATION_ID

Destination: SO_LINE_DETAILS.WAREHOUSE_ID

INTERFACE_STATUS **VARCHAR2(1000)**

This column is used internally to store information about any invalid data.

ERROR_FLAG**VARCHAR2(1)**

This column is used internally; **Y** means an error occurred when importing this header.

REQUIRED_FOR_REVENUE**VARCHAR2(1)**

This column is for internal use only.

SO_PRICE_ADJUSTMENTS_INTERFACE

The following graphic summarizes the SO_PRICE_ADJUSTMENTS_INTERFACE table. Refer to the Using IDs section on page 6 – 66, and the Required Fields for Booked Orders section on page 6 – 71 for more details.

Database columns not included in this table are for internal use only. Each column is described in detail below starting on page 6 – 125.

SO_PRICE_ADJUSTMENTS_INTERFACE Column Name	Type	Required (C indicates Condi- tionally Required)	Condi- tionally Required for Booking	Derived	Optional
CREATION_DATE	Date	✓			
CREATED_BY	Number	✓			
LAST_UPDATE_DATE	Date	✓			
LAST_UPDATED_BY	Number	✓			
LAST_UPDATE_LOGIN	Number				✓
PROGRAM_APPLICATION_ID	Number				✓
PROGRAM_ID	Number				✓
PROGRAM_UPDATE_DATE	Date				✓
REQUEST_ID	Number				✓
ORDER_SOURCE_ID	Number	✓			
ORIGINAL_SYSTEM_REFERENCE	Var- char2(50)	✓			

Table 6 – 16 SO_PRICE_ADJUSTMENTS_INTERFACE Table (Page 1 of 2)

SO_PRICE_ ADJUSTMENTS_ INTERFACE Column Name	Type	Required (C indicates Condi- tionally Required)	Condi- tionally Required for Booking	Derived	Optional
ORIGINAL_SYSTEM_	Var-				✓
LINE_REFERENCE	char2(50)				
DISCOUNT_NAME	Var-	C			
	char2(30)				
DISCOUNT_ID	Number	C			
PERCENT	Number	✓			
PRICING_CONTEXT	Var-				✓
	char2(30)				
PRICING_ATTRIBUTE1 – 15	Var-				✓
	char2(150)				
CONTEXT	Var-				✓
	char2(30)				
ATTRIBUTE1 – 15	Var-				✓
	char2(150)				
DISCOUNT_LINE_ID	Number				✓
OPERATION_CODE	Var-				✓
	char2(30)				

Table 6 – 16 SO_PRICE_ADJUSTMENTS_INTERFACE Table (Page 2 of 2)

CREATION_DATE Not Null **DATE**

Enter the date that you originally entered the data into your feeder system.

Validation: Standard Date Validation

Destination: SO_LINES.CREATION_DATE

CREATED_BY Not Null **NUMBER**

Enter an identification number that you can use to identify the user who created the record.

Validation: None

Destination: SO_LINES.CREATED_BY

LAST_UPDATE_DATE	Not Null	DATE
-------------------------	----------	-------------

Enter the current date that you are entering data into your feeder system.

Validation: Standard Date Validation

Destination: SO_LINES.LAST_UPDATE_DATE

LAST_UPDATED_BY	Not Null	NUMBER
------------------------	----------	---------------

Enter an identification number that you can use to identify the user who created or who most recently modified the record.

Validation: None

Destination: SO_LINES.LAST_UPDATE_BY

LAST_UPDATE_LOGIN		NUMBER
--------------------------	--	---------------

Enter an identification number that you can use to identify the feeder system that is supplying Oracle Order Entry with this data.

Validation: None

Destination: SO_LINES.LAST_UPDATE_LOGIN

PROGRAM_APPLICATION_ID		NUMBER
-------------------------------	--	---------------

This column is for internal use only.

PROGRAM_ID		NUMBER
-------------------	--	---------------

This column is for internal use only.

PROGRAM_UPDATE_DATE		DATE
----------------------------	--	-------------

This column is for internal use only.

REQUEST_ID		NUMBER
-------------------	--	---------------

This column is for internal use only. It is populated with the concurrent manager request ID for the run of the interface each time you submit the OrderImport program. In order for records to be processed, their request ID must be null or equal to the request ID being processed.

ORDER_SOURCE_ID	Not Null	NUMBER
------------------------	----------	---------------

Enter the order import source ID. This tells OrderImport whether to use IDs or the name columns and is used to determine which records in the interface table to process.

Validation: SO_ORDER_SOURCES

Destination: None

ORIGINAL_SYSTEM_REFERENCE	Not Null	VARCHAR2(50)
----------------------------------	----------	---------------------

Enter the order number or ID from your feeder system.

Validation: Matches SO_HEADERS_INTERFACE.ORIGINAL_SYSTEM_REFERENCE

Destination: SO_LINES.ORIGINAL_SYSTEM_REFERENCE

ORIGINAL_SYSTEM_LINE_REFERENCE	VARCHAR2(50)
---------------------------------------	---------------------

If the discount applies to the order line, enter the line ID from your feeder system. If this is an order level discount, leave this column null.

Validation: Matches
SO_LINES_INTERFACE.ORIGINAL_SYSTEM_LINE_REFERENCE

Destination: SO_PRICE_ADJUSTMENTS.ORIGINAL_SYSTEM_LINE_REFERENCE

DISCOUNT_NAME	Conditionally Not Null	VARCHAR2(30)
----------------------	------------------------	---------------------

Enter the name of the discount that was applied to this order line if the selling price is different from the list price. If you are using IDs, you must enter the DISCOUNT_ID. If you are not using IDs, you must enter the DISCOUNT_NAME.

Validation: SO_DISCOUNTS.NAME

Destination: insert a record into
SO_PRICE_ADJUSTMENTS.DISCOUNT_NAME
referencing SO_DISCOUNT.DISCOUNT_NAME,
where SO_LINES_INTERFACE.DISCOUNT =
SO_DISCOUNTS.NAME

DISCOUNT_ID	Conditionally Not Null	NUMBER
--------------------	------------------------	---------------

Enter the discount ID which matches the discount name you want to use for this order line.

Validation:	SO_DISCOUNTS.DISCOUNT_ID
Destination:	SO_PRICE_ADJUSTMENTS.DISCOUNT_ID
Validation:	SO_DISCOUNT.DISCOUNT_LINES_ID
Destination:	SO_PRICE_ADJUSTMENTS.DISCOUNT_LINES_ID

PERCENT	(Required)	NUMBER
----------------	------------	---------------

Enter the percent for the discount. This percent must match the percent defined on the discount if the discount is not overridable. If you entered N in the SO_LINES_INTERFACE.CALCULATE_PRICE column, the resulting amount of this percent discount on the line must equal the difference between the list price and selling price.

Validation:	SO_DISCOUNTS.PERCENT
Destination:	SO_PRICE_ADJUSTMENTS.PERCENT

PRICING_CONTEXT	VARCHAR2(30)
------------------------	---------------------

Enter the context for your pricing descriptive flexfield attributes if you have enabled a context sensitive pricing descriptive flexfield on SO_PRICE_ADJUSTMENTS.

Validation:	None
Destination:	SO_PRICE_ADJUSTMENTS.PRICING_CONTEXT

PRICING_ATTRIBUTE1 – 15	VARCHAR2(150)
--------------------------------	----------------------

Enter any information you wish to pass to the pricing descriptive flexfield columns in SO_PRICE_ADJUSTMENTS.

Validation:	None
Destination:	SO_PRICE_ADJUSTMENTS.PRICING_ATTRIBUTE1 – 15

CONTEXT	VARCHAR2(30)
----------------	---------------------

Enter the context for your descriptive flexfield attributes if you have enabled a context sensitive descriptive flexfield on SO_PRICE_ADJUSTMENTS.

Validation: None

Destination: SO_PRICE_ADJUSTMENTS.CONTEXT

ATTRIBUTE1 – 15	VARCHAR2(150)
------------------------	----------------------

Enter any information you wish to pass to the descriptive flexfield columns in SO_PRICE_ADJUSTMENTS.

Validation: None

Destination: SO_PRICE_ADJUSTMENTS.ATTRIBUTE1 – 15

AUTOMATIC_FLAG	VARCHAR2(1)
-----------------------	--------------------

This column is for internal use only.

DISCOUNT_LINE_ID	NUMBER
-------------------------	---------------

If this applies to a specific item, order type, agreement, discount, and so on, enter the discount line ID which matches the discount name you want to use for this order line.

INTERFACE_STATUS	VARCHAR2(1000)
-------------------------	-----------------------

This column is used internally to store information about any invalid data.

ERROR_FLAG	VARCHAR2(1)
-------------------	--------------------

This column is used internally; **Y** means an error occurred when importing this header.

OPERATION_CODE	VARCHAR2(30)
-----------------------	---------------------

Enter **UPDATE** or **DELETE** for existing price adjustments, or enter **INSERT** for new imported price adjustments. You can also leave this column null for new imported price adjustments.

SO_SALES_CREDITS_INTERFACE

The following graphic summarizes the SO_SALES_CREDITS_INTERFACE table. Database columns with a check under Conditionally Required for Booking are optional if you are importing an order with an Entry Status of **Entered**. If you are importing an order with an Entry Status of **Booked**, you must enter the ID column if you are using IDs, or the name column if you are not using IDs. Refer to the Using IDs section on page 6 – 66, and the Required Fields for Booked Orders section on page 6 – 71 for more details.

Database columns not included in this table are for internal use only. Each column is described in detail below starting on page 6 – 125.

SO_SALES_CREDITS_INTERFACE Column Name	Type	Required (C indicates Condi- tionally Required)	Condi- tionally Required for Booking	Derived	Optional
CREATION_DATE	Date	✓			
CREATED_BY	Number	✓			
LAST_UPDATE_DATE	Date	✓			
LAST_UPDATED_BY	Number	✓			
LAST_UPDATE_LOGIN	Number				✓
PROGRAM_APPLICATION_ID	Number				✓
PROGRAM_ID	Number				✓
PROGRAM_UPDATE_DATE	Date				✓
REQUEST_ID	Number				✓
ORIGINAL_SYSTEM_REFERENCE	Var- char2(50)	✓			
ORIGINAL_SYSTEM_LINE_REFERENCE	Var- char2(50)				✓
ORDER_SOURCE_ID	Number	✓			
SALESREP_NAME	Var- char2(30)		✓		
SALESREP_ID	Number		✓		

Table 6 – 17 SO_SALES_CREDITS_INTERFACE Table (Page 1 of 2)

SO_SALES_CREDITS_INTERFACE Column Name	Type	Required (C indicates Condi- tionally Required)	Condi- tionally Required for Booking	Derived	Optional
SALES_CREDIT_TYPE	Var- char2(30)	C			
SALES_CREDIT_TYPE_ID	Number	C			
QUOTA_FLAG	Varchar2(1)			✓	
PERCENT	Number	✓			
CONTEXT	Var- char2(30)				✓
ATTRIBUTE1 – 15	Var- char2(150)				✓
OPERATION_CODE	Var- char2(30)				✓

Table 6 – 17 SO_SALES_CREDITS_INTERFACE Table (Page 2 of 2)

CREATION_DATE Not Null **DATE**

Enter the date that you originally entered the data into your feeder system.

Validation: Standard Date Validation

Destination: SO_LINE.CREATION_DATE

CREATED_BY Not Null **NUMBER**

Enter an identification number that you can use to identify the feeder system that is supplying Oracle Order Entry with this data.

Validation: None

Destination: SO_SALES_CREDITS.CREATED_BY

LAST_UPDATE_DATE Not Null **DATE**

Enter the date that you originally entered the data into your feeder system.

Validation: Standard Date Validation

Destination: SO_SALES_CREDITS.LAST_UPDATE_DATE

LAST_UPDATED_BY Not Null **NUMBER**

Enter an identification number that you can use to identify the feeder system that is supplying Oracle Order Entry with this data.

Validation: None

Destination: SO_SALES_CREDITS.LAST_UPDATE_BY

LAST_UPDATE_LOGIN **NUMBER**

Enter an identification number that you can use to identify the feeder system that is supplying Oracle Order Entry with this data.

Validation: None

Destination: SO_SALES_CREDITS.LAST_UPDATE_LOGIN

PROGRAM_APPLICATION_ID **NUMBER(15)**

This column is for internal use only.

PROGRAM_ID **NUMBER(15)**

This column is for internal use only.

PROGRAM_UPDATE_DATE **DATE**

This column is for internal use only.

REQUEST_ID **NUMBER**

This column is for internal use only. It is populated with the concurrent manager request ID for the run of the interface each time you submit the OrderImport program. In order for records to be processed, their request ID must be null or equal to the request ID being processed.

ORIGINAL_
SYSTEM_REFERENCE Not Null **VARCHAR2(50)**

Enter the order number or ID from the feeder system.

Validation: Must match
SO_HEADERS_INTERFACE.ORIGINAL_
SYSTEM_REFERENCE and

SO_LINES_INTERFACE.ORIGINAL_SYSTEM_
REFERENCE

Destination: None

**ORIGINAL_SYSTEM_
LINE_REFERENCE** **VARCHAR2(50)**

If the sales credit applies to the order line, enter the line ID from your feeder system. If this is an order level sales credit, leave this column null.

Validation: Matches
SO_LINES_INTERFACE.ORIGINAL_SYSTEM_
LINE_REFERENCE

Destination: SO_SALES_CREDITS.ORIGINAL_SYSTEM_
LINE_REFERENCE

ORDER_SOURCE_ID Not Null **NUMBER**

Enter the order import source ID. This tells OrderImport whether to use IDs or the name columns.

Validation: SO_ORDER_SOURCES

Destination: None – used to determine which records, in the interface table, to process and whether IDs are required.

LINE_NUMBER **NUMBER**

This column is for internal use only.

LINE_TYPE **VARCHAR2(30)**

This column is for internal use only.

SALESREP_NAME **VARCHAR2(30)**

Enter the name of the salesperson to whom you are assigning the sales credit. If you want multiple credits, you must have multiple records in the sales_credits_interface table. The total quota credit will always have to equal 100% for each line and for the header.

Validation: Must be Active in RA_SALESREPS.NAME for current orders, need only exist for complete orders

Destination: inserts RA_SALESREP.SALESREP_ID into
SO_SALES_CREDITS.SALESREP_ID where
SO_SALES_CREDITS_INTERFACE.SALESREP_
NAME = RA_SALESREPS.NAME

<u>SALESREP_ID</u>	<u>NUMBER</u>
--------------------	---------------

Enter the salesrep id that matches the SALESREP_NAME you entered.

Validation: RA_SALESREP.SALESREP_ID

Destination: SO_SALES_CREDITS.SALESREP_ID

<u>SALES_CREDIT_TYPE</u>	Conditionally Not Null	<u>VARCHAR2(30)</u>
--------------------------	------------------------	---------------------

Enter the name of the sales credit type you are assigning to the salesrep (such as **Quota Sales Credit** or **Nonquota Sales Credit**). If you are using IDs, you must enter the SALES_CREDIT_TYPE_ID. If you are not using IDs, you must enter the SALES_CREDIT_TYPE, and if your SALES_CREDIT_TYPE is not unique, you must also enter the SALES_CREDIT_TYPE_ID.

Validation: SO_SALES_CREDIT_TYPES.NAME

Destination: inserts
SO_SALES_CREDIT_TYPES.SALES_CREDIT_
TYPE_ID into SO_SALES_CREDITS where
SO_SALES_CREDITS_INTERFACE.SALES_
CREDIT_TYPE =
SO_SALES_CREDIT_TYPES.NAME

<u>SALES_ CREDIT_TYPE_ID</u>	Conditionally Not Null	<u>NUMBER</u>
----------------------------------	------------------------	---------------

Enter the SALES_CREDIT_TYPE_ID for the SALES_CREDIT_TYPE you entered. If you are using IDs, you must enter the SALES_CREDIT_TYPE_ID. If you are not using IDs, you must enter the SALES_CREDIT_TYPE, and if your SALES_CREDIT_TYPE is not unique, you must also enter the SALES_CREDIT_TYPE_ID.

Validation: SO_SALES_CREDIT_TYPES.SALES_CREDIT_
TYPE_ID

Destination: SO_SALES_CREDITS.SALE_CREDIT_TYPE_ID

QUOTA_FLAG**VARCHAR2(1)**

Leave this field null, it is derived from the SALES_CREDIT_TYPE column.

PERCENT

Not Null

NUMBER

Enter the percent of the sales credit type you are assigning to the salesperson. The total quota credit will always have to equal 100% for each line and for the header.

Validation: The total must be less than or equal to 100 for quota type sales credits. Nonquota sales credits can exceed 100.

For header level sales credits:

```
select count(*) into :SO_SALES_CREDITS_ERRORS
from SO_SALES_CREDITS_INTERFACE
where NOT EXISTS
  ((select 'X' from dual
   where 100 = (select sum(SO_SALES_CREDITS.PERCENT)
                from SO_SALES_CREDITS, SO_HEADERS
                where SO_SALES_CREDITS_INTERFACE.
                     ORIGINAL_SYSTEM_REFERENCE
                     = SO_HEADERS.ORIGINAL_SYSTEM_REFERENCE
                     and SO_SALES_CREDITS_INTERFACE.
                     ORDER_SOURCE_ID
                     = SO_HEADERS.ORIGINAL_SYSTEM_SOURCE_CODE
                     and SO_HEADERS.HEADER_ID
                     = SO_SALES_CREDITS.HEADER_ID
                     and SO_HEADERS.S1 = 1 /** Booked **/
                     and SO_SALES_CREDITS.LINE_ID IS NULL
                     and 'Y' = (select QUOTA_FLAG
                                from SO_SALES_CREDIT_TYPES
                                where SO_SALES_CREDIT_TYPES.SALES_CREDIT_TYPE_ID
                                = SO_SALES_CREDITS.SALES_CREDIT_TYPE_ID)))
union ((select 'X' from dual
 where 100 >= (select nvl(sum(SO_SALES_CREDITS.PERCENT), 0)
                from SO_SALES_CREDITS, SO_HEADERS
                where SO_SALES_CREDITS_INTERFACE.
                     ORIGINAL_SYSTEM_REFERENCE
                     = SO_HEADERS.ORIGINAL_SYSTEM_REFERENCE
                     and SO_SALES_CREDITS_INTERFACE.ORDER_SOURCE_ID
                     = SO_HEADERS.ORIGINAL_SYSTEM_SOURCE_CODE
                     and SO_HEADERS.HEADER_ID
                     = SO_SALES_CREDITS.HEADER_ID
```

```

and SO_HEADERS.S1 != 1 /** Not booked **/
and SO_SALES_CREDITS.LINE_ID IS NULL
and 'Y' = (select QUOTA_FLAG
from SO_SALES_CREDIT_TYPES
where SO_SALES_CREDIT_TYPES.SALES_CREDIT_TYPE_ID
= SO_SALES_CREDITS.SALES_CREDIT_TYPE_ID))))

```

For line level sales credits:

The same as above except SO_SALES_CREDITS.LINE_ID is NOT NULL instead of NULL.

Destination: SO_SALES_CREDITS.PERCENT

USE_IDS_FLAG **VARCHAR2(1)**

This column is for internal use only.

CONTEXT **VARCHAR2(30)**

Enter the context for your descriptive flexfield information if you have enabled a context sensitive descriptive flexfield on SO_SALES_CREDITS.

Validation: None

Destination: SO_SALES_CREDITS.CONTEXT

ATTRIBUTE1 – 15 **VARCHAR2(150)**

Enter any information you wish to pass to the descriptive flexfield column in so_sales_credits.

Validation: None

Destination: SO_SALES_CREDITS.ATTRIBUTE1 – 10

INTERFACE_STATUS **VARCHAR2(1000)**

This column is used internally to store information about any invalid data.

ERROR_FLAG **VARCHAR2(1)**

This column is used internally; if a Y is generated it means an error occurred when importing this header.

OPERATION_CODE**VARCHAR2(30)**

Enter **UPDATE** or **DELETE** for existing sales credits, or enter **INSERT** for new imported sales credits. You can also leave this column null for new imported sales credits.

Oracle Payables Applications Open Interfaces

This chapter contains information about the following Oracle Payables Applications open interfaces:

- Integrating Payment Reconciliation Information with Oracle Payables or Oracle Government Payables
- Importing Invoice Information into Oracle Payables or Oracle Government Payables
- Integrating Purchase Order Information with Oracle Payables or Oracle Government Payables

Integrating Payment Reconciliation Information with Oracle Payables or Oracle Government Payables

AutoClear is the Oracle Payables or Oracle Government Payables feature you use to perform automatic payment reconciliation for the payments in your Oracle Payables application. AutoClear allows you to quickly and easily compare the payment clearing information from your large-volume banks with the payment information in your Oracle Payables application and update your payment history to reflect the transactions your banks have processed.

You reconcile your payment history with the payment clearing information from your banks so you can respond quickly and accurately to vendor payment inquiries. And, you reconcile your payment history regularly so you can spot potential payment problems. For example, if a payment remains uncleared for several months, it may be lost in the mail.

Your Oracle Payables application also supports manual reconciliation for your payments.



Reconcile Payments

(*Oracle Payables Reference Manual* or *Oracle Government Payables Reference Manual*)

Alternatively, you can use Oracle Cash Management to reconcile bank statements with subsidiary ledger transactions from Oracle Applications and Oracle Receivables. You should reconcile your payments using either the Reconcile Payments form or AutoClear program in Oracle Applications, or the Automatic and Manual Bank Reconciliation forms in Oracle Cash Management. We suggest that you do not use both the methods of reconciliation concurrently. If you choose to use the Oracle Cash Management System to perform your reconciliation, all your payments will be cleared and matched to individual bank statement lines. You cannot use Oracle Cash Management to reconcile your payments if you have enabled the Automatic Offsets system option in Oracle Payables or the Fund Accounting system option in Oracle Government Payables.

You can use Oracle Cash Management to automatically or manually reconcile your payments and receipts to a bank statement. Oracle Cash Management records the appropriate accounting for you, including cash clearing, gains and losses, even bank charges and errors. You define the accounts Oracle Cash Management uses for cash clearing, cash, bank errors, bank charges, and gains and losses in the Set Up Bank Information form in Oracle Applications. You enable integration

with Oracle Cash Management with the Allow Cash Clearing Account system option.

Oracle Cash Management is a new application. Please refer to the Oracle Cash Management Users Guide for more details.



Additional Information:

Oracle Cash Management User's Guide

Basic Integration Needs

AutoClear provides you with the features you need to satisfy the following basic integration needs:

- Import bank information from tapes or diskettes provided by your bank
- Validate all imported reconciliation data against your bank data
- Quickly update your payment history for payments large-volume bank accounts with cleared payment information
- Reconcile multiple accounts at the same time
- Define any necessary reconciliation data that your bank does not provide
- Allow on-line correction of reconciliation exceptions that AutoClear identifies during the AutoClear process
- Report on the results of your automatic payment reconciliation

Major Features

Automatic Payment Reconciliation

Your Oracle Payables application provides you with the features you need to automatically update all cleared payments with the information your bank provides. Using AutoClear, you can import information from your bank into your Oracle Payables application and use the imported information to update the payments that cleared.

Validation and Error Checking

AutoClear verifies that all the payment information you import from a bank file is valid in your Oracle Payables application. It also checks the imported payment information for errors based on the bank account error checking groups you define. The error checking groups you define correspond to the error checking controls that your bank uses.

For example, your bank may provide the total number and amount of all cleared checks in its file. You can define an error checking group that counts the total number of records in the file and verifies that this number matches the total provided by the bank. You can define a second error checking group that calculates the total amount for the cleared checks in the file and verifies that the sum is correct before proceeding with the remainder of the payment reconciliation.

SQL*Loader Control Files

AutoClear uses SQL*Loader to load the information from your bank file into the Oracle Payables application database. SQL*Loader uses a control file that you create to validate and format the data being loaded. You can also use the control file to provide your Oracle Payables application with required AutoClear information that your bank does not provide.

If your bank provides you with payment information for multiple bank accounts in a single reconciliation file and provides bank account information for each payment record in the file, you can reconcile multiple accounts in a single run of AutoClear. You can specify in your control file all the bank account information provided by your bank. When you run AutoClear, your Oracle Payables application updates each bank account defined in your control file.

Resolve AutoClear Exceptions Form

Use this form to correct exceptions that AutoClear identifies during reconciliation. Exceptions may be records that AutoClear does not reconcile (e.g. deposits, bank charges) or records that AutoClear could not reconcile due to differences between the payment in the bank file and the payment in your Oracle Payables application.

Reconciliation Reports

Your Oracle Payables application prints the Reconciliation Summary Report and the Payment Reconciliation Exceptions Report each time you initiate AutoClear. The Reconciliation Summary Report provides

summary information for each reconciled and unreconciled record, including the record status and the number of unreconciled records for each bank account. The Payment Reconciliation Exceptions Report provides detailed information for each payment record that AutoClear could not reconcile with your bank accounts.

In addition to these two reports, your Oracle Payables application provides you with the Payment Reconciliation Detail Report that you can submit from the Resolve AutoClear Exceptions form. You can use this report as a record of your payment reconciliation activity after using AutoClear. The Payment Reconciliation Detail Report lists, by bank account, all the payment records in the bank file that AutoClear reconciled with the payment records in your Oracle Payables application, as well as any unresolved reconciliation/exceptions. This report also provides a summary of payment reconciliation activity by bank account.

AutoClear Definitions

Error Checking Group

An Oracle Payables feature you use to verify that information you load from a reconciliation file is accurate and complete. Your Oracle Payables application compares the error checking groups you define against the error checking groups your bank uses to identify errors in the file.

For example, your bank provides a count of the total number of records in a reconciliation file, so you define an error checking group to count the number of records in the file and compare them to your bank total. During AutoClear, your Oracle Payables application verifies that the totals match, and, if they do not match, aborts AutoClear.

Mask

An Oracle Payables application feature which allows you to load information (dates or figures) in a format other than the format that your Oracle Payables application utilizes.

For example, your bank may send you dates in the format DD/MM/YY. Because your Oracle Payables application does not support this date format, you must define a mask that allows your Oracle Payables application to read dates in this format.

Reconciliation File

A file your bank sends to you on a tape or diskette that contains payment transaction information for your accounts. You load this information into your Oracle Payables application to perform automatic payment reconciliation

Record Code

A code your bank uses to describe the types of records in the payment reconciliation file they send you.

For example, a bank may use the record codes R01, R02, and R03 to represent, respectively, outstanding checks, bank-originated entries (wire transfers), and paid checks.

To use AutoClear, you define record codes in your Oracle Payables application that match the record codes your bank uses.

Transaction Code

A code your bank uses to describe the types of transactions for the records in the reconciliation file they send you.

For example, a bank may use the transaction codes T01, T02, and T03 to represent, respectively, debits, credits, and stop payments.

To use AutoClear, you define transaction codes in your Oracle Payables application that match the transaction codes your bank uses.

Using AutoClear in Oracle Payables Applications

To use AutoClear, your bank must provide payment clearing information on a tape or diskette. And, you must create a SQL*Loader control file that AutoClear uses to load your payment information from the tape or diskette into your AutoClear interface tables.

During reconciliation, AutoClear compares the information you load into the interface tables with the information in your payment tables. If the information matches, your Oracle Payables application updates the status of a payment in your Oracle Payables application based on the status of the payment clearing information sent to you by your bank.

SQL*Loader and the SQL*Loader Control File

AutoClear uses SQL*Loader to load data from your bank reconciliation file into the Oracle Payables application database. SQL*Loader uses the control file that you create to format and load the payment data in the reconciliation file into the AutoClear interface tables.

You can create your own SQL*Loader control file or you can use the sample control file **APLDRNPX.ctl** that your Oracle Payables application provides. The sample control file is located in the **bin** subdirectory of the **AP_TOP** directory tree.

Multiple Control Files

If your bank uses multiple accounts and does not provide an account number in every record, you must develop one bank control file for each account the bank uses. If you have accounts for multiple banks and you want to use AutoClear, you must develop separate control files for each bank account in each bank.

If you create multiple control files, you must run AutoClear separately for each control file. You can create a subdirectory under the **bin** directory where you can store your multiple control files. When you want to run AutoClear, you can rename the control file for the account you are reconciling as **APLDRNPX.ctl** and move it from your subdirectory to the **bin** directory.

Understanding the AutoClear Tables

AutoClear loads bank information for each record in the reconciliation file into one of the following interface tables:

- **AP_RECON_DETAILS**
- **AP_RECON_SUMMARIES**

AutoClear loads information for each payment transaction in the file into **AP_RECON_DETAILS**. AutoClear loads all summary information, which it uses for error checking, into **AP_RECON_SUMMARIES**.

The interface tables are organized by columns which your Oracle Payables application uses to categorize and store specific bank and payment transaction information. For example, bank record codes from the reconciliation file are stored in the **AP_RECON_DETAILS.RECORD_CODE** column.

Additional AutoClear Tables

In addition to the interface tables, AutoClear uses the following tables for reconciling payments:

- AP_CHECKS_ALL
- AP_RECON_CHECK_RESULTS
- AP_RECON_ERROR_CHK_GROUPS
- AP_RECON_ERROR_CHK_GROUP_LINES
- AP_RECON_LOCKS
- AP_RECON_RECORD_CODES
- AP_RECON_RECORD_SPECS
- AP_RECON_TRANSACTION_CODES

Table Descriptions

The following section describes the columns in the AutoClear interface tables, AP_RECON_DETAILS and APRECON_SUMMARIES, and indicates the columns that require values for automatic payment reconciliation. For a complete description of the AutoClear interface tables, please consult the *Oracle Payables Applications Technical Reference Manual*.



Table Definitions

(*Oracle Payables Applications Technical Reference Manual*)

AP_RECON_DETAILS

Column Name	Null	Type	Comments
BANK_NAME	NOT NULL	VARCHAR2(30)	
BANK_ACCOUNT_NUM	NOT NULL	VARCHAR2(30)	
RECORD_CODE	NOT NULL	VARCHAR2(30)	
LAST_UPDATE_DATE	NOT NULL	DATE	Who column
LAST_UPDATED_BY	NOT NULL	NUMBER(15)	Who column
BANK_ACCOUNT_ID		NUMBER(15)	Null

Table 7 – 1 AP_RECON_DETAILS (Page 1 of 2)

Column Name	Null	Type	Comments
TRANSACTION_CODE		VARCHAR2(30)	Required
SERIAL_NUMBER		NUMBER(25)	Required, Payment number
CLEARED_DATE		DATE	Required
AMOUNT		NUMBER	Required Cleared amount
CURRENCY_CODE		VARCHAR2(15)	
RECORD_STATUS		VARCHAR2(30)	Null, Payables updates with AutoClear status
LAST_UPDATE_LOGIN		NUMBER(15)	Who column
CREATION_DATE		DATE	Who column
CREATED_BY		NUMBER(15)	Who column
TREASURY_PAY_DATE		DATE	Not used
TREASURY_PAY_NUMBER		NUMBER(15)	Not used

Table 7 – 1 AP_RECON_DETAILS (Page 2 of 2)

AP_RECON_SUMMARIES

Column Name	Null	Type	Comments
RECORD_CODE	NOT NULL	VARCHAR2(30)	
BANK_ACCOUNT_ID		NUMBER(15)	Null
BANK_NAME	NOT NULL	VARCHAR2(30)	
BANK_ACCOUNT_NUM	NOT NULL	VARCHAR2(30)	
LAST_UPDATE_DATE	NOT NULL	DATE	Who column
LAST_UPDATED_BY	NOT NULL	NUMBER(15)	Who column
ERROR_CHECK1-10		NUMBER(19,2)	Error checking in- formation

Table 7 – 2 AP_RECON_SUMMARIES (Page 1 of 1)

Setting Up AutoClear

The following list summarizes the procedure for setting up AutoClear in your Oracle Payables application. The rest of the section explains each step in detail.

Setup Checklist

- ☐ 1. Set Up Your Bank Record Codes
- ☐ 2. Set Up Your Bank Transaction Codes
- ☐ 3. Set Up Your Bank File Specifications
- ☐ 4. Read SQL*Loader Documentation (Optional)
- ☐ 5. Create/Modify SQL*Loader Control File

Setup Steps

Complete the following steps to set up AutoClear:

Step 1 **Set Up Your Bank Record Codes**

You define bank record codes using the Define Bank Codes form. Your record codes must match the codes your bank sends you in the reconciliation file. For each record code, you define a record type (Detail, Summary, or Header), which identifies the type of record that your bank sends you. Your Oracle Payables application uses Summary record codes for identifying error checking information and Detail record codes for identifying the transactions in a reconciliation file.



Attention: Your Oracle Payables application does not make use of records with Header record codes.

For example, your bank may send a file with a header record that contains the name of your bank and the account number, followed by a summary record that lists the number and of payments and total in the file, and detail records with information about the individual payments. You define a Header and a Summary record code and one Detail record code for each detail record.

If your bank does not use record codes, you need to create at least one Detail record code so that you can specify it in your control file (see Step 5).



Define Bank Codes
(*Oracle Payables Reference Manual* or *Oracle Government Payables Reference Manual*)

Your Oracle Payables application stores your record codes in AP_RECON_RECORD_CODES.

Step 2 Set Up Your Bank Transaction Codes

After you define your bank record codes, you define bank transaction codes in the same form. Your transaction codes must match the codes your bank sends you in the reconciliation file. For each transaction code, you define an action (Debit, Credit, or Stop), which identifies the type of transaction performed on a record.



Attention: Your Oracle Payables application does not reconcile records with Credit or Stop transaction codes and reports the records as exceptions.

If your bank does not use transaction codes, you need to create at least one Debit transaction code so that you can specify it in your control file (see Step 5).

Your Oracle Payables application stores your transaction codes in AP_RECON_TRANSACTION_CODES.

Step 3 Set Up Your Bank File Specifications

After you define your bank transaction codes, you use the Define Bank File Specifications form to describe the format of the bank reconciliation file your bank sends you. You describe your bank file format so your Oracle Payables application knows where to look in your bank file for the information it uses to reconcile your bank account with your Oracle Payables application payment history. For each column in the AutoClear interface tables, you specify the type of information you are going to load and the format of the information. For example, in the AMOUNT column, you may specify that positions 1 through 10 contain the amount of a payment.

Specify an amount mask to convert AMOUNT to the amount format that your Oracle Payables application supports. You do not need to specify a date mask, separation method, or separation character for any

of the columns you define. Your Oracle Payables application uses your SQL*Loader control file to determine the formats for these values.

You also use this form to define the error checking groups that your bank uses. AutoClear uses these error checking groups to validate the payment information that it loads from the reconciliation file provided by your bank. You can define one error checking group for each ERROR_CHECK column in the AP_RECON_SUMMARIES table. In addition, you can define multiple error checking lines for each group.

You can define two types of error checking group, Sum and Count. Sum error checking groups validates the sum of the records you specify against a sum that your bank provides in the bank reconciliation file. Count error checking groups validate the total number of a specified type of record against the total that your bank provides.



Define Bank File Specifications

(Oracle Payables Reference Manual or Oracle Government Payables Reference Manual)



Attention: Currently you need to define a SQL*Loader control file as well as set up bank file specifications. In a future release, your Oracle Payables application will use the bank file specifications that you define in the Define Bank File Specifications form to automatically create a SQL*Loader control file.

Your Oracle Payables application stores your bank file specifications in AP_RECON_RECORD_SPECS and your error checking groups and lines in AP_RECON_ERROR_CHK_GROUPS and AP_RECON_ERROR_CHK_GROUP_LINES.

Step 4 **Read SQL*Loader Documentation (Optional)**

Before you attempt to create a control file or modify the sample control file that your Oracle Payables application provides, you should familiarize yourself with SQL*Loader. You use SQL*Loader to load information from the reconciliation file your bank gives you into your Oracle Payables application database.



Additional Information: SQL*Loader
(ORACLE7 Server Utilities Guide)

Step 5 **Create/Modify SQL*Loader Control File**

To use AutoClear, you must create a SQL*Loader control file. SQL*Loader uses this file to format the data in your bank file so that the data loads successfully into the columns in the AutoClear interface

tables. You can also modify the sample control file, APLDRNPX.ctf, provided by your Oracle Payables application. If you create your own bank control file, you must name the file APLDRNPX.ctf.



Suggestion: Before you create or modify your APLDRNPX.ctf file, we recommend that you make a copy of the original sample file and use the copy to make your modifications. Your original file then serves as a back-up and reference copy. The file that you modify must be named APLDRNPX.ctf, so you will need to rename the original file (e.g. APLDRNPX.ctf.old).

The following figure provides an example of the sample control file your Oracle Payables application provides for automatic payment reconciliation:

Figure 7 – 1 Sample SQL*Loader control file

```
LOAD DATA
APPEND
INTO TABLE ap_recon_details
WHEN (1:1) = 'H'
  (bank_name CONSTANT "Bank Name" ,
  bank_account_num CONSTANT "Bank Account Number" ,
  record_code POSITION (1:1) CHAR ,
  last_update_date CONSTANT "01-JAN-89" ,
  last_updated_by CONSTANT "1"
  )
INTO TABLE ap_recon_details
WHEN (1:1) = 'R'
  (bank_name CONSTANT "Bank Name" ,
  bank_account_num CONSTANT "Bank Account Number" ,
  record_code POSITION (1:1) CHAR ,
  transaction_code POSITION (14:16) CHAR ,
  last_update_date CONSTANT "01-JAN-89" ,
  last_updated_by CONSTANT "1" ,
  serial_number POSITION (4:13) INTEGER EXTERNAL ,
  cleared_date POSITION (33:38) DATE "DDMMYY" ,
  amount POSITION (23:32) DECIMAL EXTERNAL
  )
INTO TABLE ap_recon_summaries
WHEN (1:1) = 'T'
  (bank_name CONSTANT "Bank Name" ,
  bank_account_num CONSTANT "Bank Account Number" ,
  record_code POSITION (1:1) CHAR ,
  last_update_date CONSTANT "01-JAN-89" ,
  last_updated_by CONSTANT "1" ,
  error_check1 POSITION (4:9) INTEGER EXTERNAL ,
  error_check2 POSITION (11:21) INTEGER EXTERNAL ,
  error_check3 POSITION (24:29) INTEGER EXTERNAL ,
  error_check4 POSITION (31:41) INTEGER EXTERNAL
  )
```

The sample file assumes:

- Your bank sends bank account information in a fixed file format
- Your bank does not store the bank account number in all records

If your bank sends information in a flexible file format, refer to SQL*Loader documentation for assistance in developing the APLDRNPX.ctf file.



Additional Information: SQL*Loader
(*ORACLE7 Server Utilities Guide*)

In order for SQL*Loader to load your bank reconciliation file properly, each logical record your bank sends must end with a carriage return. You will receive a SQL*Loader error message when you initiate AutoClear if each record does not end with a carriage return.

Required Reconciliation Information

You must specify certain information in your APLDRNPX.ctl file, such as bank name and bank account number. You also need to specify positional numbers, record codes, and transaction codes for your bank account. Enter the exact positions of your payment details and do not zero pad any of the numbers you specify.



Attention: In addition to required columns, you must populate all NOT NULL columns in the reconciliation interface tables.

The minimum information your bank must provide so you can use AutoClear is the serial number, cleared date, and amount for each bank record. If you do not receive this information, your Oracle Payables application is unable to reconcile your bank account with your payment history.

You must specify any information that your Oracle Payables application requires which your bank does not provide. For example, if your bank does not provide transaction codes (i.e. debit, credit, or stop), you must create at least one transaction code and specify it in your APLDRNPX.ctl file.



Warning: Do not enter a bank account ID through SQL*Loader. Your Oracle Payables application enters this information automatically into the interface tables during AutoClear.

The sample APLDRNPX.ctl file does not include information about bank account currency. If you know that all your transactions for a bank account will always be in a particular currency, you can specify this currency in your APLDRNPX.ctl file. If you do not specify this information in your APLDRNPX.ctl file, AutoClear automatically assigns the currency of the bank account you defined in the Set Up Bank Information form to each transaction. If you specify information in the control file that you have already defined in your Oracle Payables application, such as currency code, bank name, or account

number, you must specify the information using the exact names you defined in your application.



Define Currency

Set Up Bank Information

(Oracle Payables Reference Manual or Oracle Government Payables Reference Manual)

Example You have a bank account with the Bank of San Francisco. The currency for your bank account is USD and the bank account number is 060600135651. You use the Set Up Bank Information form to enter bank account number 06060013565 for Bank of San Francisco.

The bank sends you a reconciliation file with the following specifications:

- SERIAL_NUMBER in positions 4–13
- AMOUNT in positions 17–32
- CLEARED_DATE in positions 33–38
- RECORD_CODE (Header = H, Summary = T, Detail = R) in positions 1–3
- TRANSACTION_CODE (Debit = 57, Bank Debit = 58, Credit = 83) in positions 14–16
- Summary line contains number and sum of Debit and Credit records

Figure 7 – 2 illustrates how the reconciliation file from your bank might look.

Figure 7 – 2 Sample bank reconciliation file

H1 060600135651					
T1 000014 00000095464 000002 00000061472					
R01000025000105700000000000001630012194	4				
R01000025000205700000000000002510012194	4				
R0100002500030570000000000000400012194	4				
R010000250004057000000000000010942010794	4				
R010000250005057000000000000020824010794	4				
R010000250006057000000000000008665011194	4				
R010000250007057000000000000013101011194	4				
R01000025000805700000000000005600011194	4				
R01000025000905700000000000003656011494	4				
R01000025001005700000000000007600011194	4				
R01560455023107900000000000043072011994	2				
R01560455023107900000000000018400012094	2				
R01000025010005700000000000009600012094	4				
R01000025010105700000000000009345012094	4				
R01560455023105800000000000001115010894	4				
R0156045502310580000000000000476011994	4				
1-3 Record Code	4-13 Serial Number	14-16 Transaction Code	17-32 Amount	33-38 Date	39-42 Other Bank Code

When you create your APLDRNPX.ctl file, you enter Bank of San Francisco and 06060013565 for the bank name and bank account number. You also specify USD as the currency.


 **Attention:** You must enter your bank name and accounting number using the exact names you defined in your application. In this example, if you were to enter Bank of San Fran, or 06-0600-13565, you would receive an error message when you initiate AutoClear.

Figure 7 – 3 shows the APLDRNPX.ctl file modified for Bank of San Francisco, account 06060013565, and the file specifications listed.

**Figure 7 – 3 Sample of
a modified SQL*Loader
control file**

```

LOAD DATA
APPEND
INTO TABLE ap_recon_details
WHEN (1:1) = 'H'
  (bank_name CONSTANT "Bank of San Francisco" ,
  bank_account_num CONSTANT "06060013565" ,
  record_code POSITION (1:1) CHAR ,
  last_update_date CONSTANT "01-FEB-94" ,
  last_updated_by CONSTANT "1"
  )
INTO TABLE ap_recon_details
WHEN (1:1) = 'R'
  (bank_name CONSTANT "Bank of San Francisco" ,
  bank_account_num CONSTANT "06060013565" ,
  record_code POSITION (1:1) CHAR ,
  transaction_code POSITION (14:16) CHAR ,
  last_update_date CONSTANT "01-FEB-94" ,
  last_updated_by CONSTANT "1" ,
  serial_number POSITION (4:13) INTEGER EXTERNAL ,
  cleared_date POSITION (33:38) DATE "DDMMYY" ,
  amount POSITION (17:32) DECIMAL EXTERNAL ,
  currency_code CONSTANT "USD"
  )
INTO TABLE ap_recon_summaries
WHEN (1:1) = 'T'
  (bank_name CONSTANT "Bank of San Francisco" ,
  bank_account_num CONSTANT "06060013565" ,
  record_code POSITION (1:1) CHAR ,
  last_update_date CONSTANT "01-FEB-94" ,
  last_updated_by CONSTANT "1" ,
  error_check1 POSITION (4:9) INTEGER EXTERNAL ,
  error_check2 POSITION (11:21) DECIMAL EXTERNAL ,
  error_check3 POSITION (23:28) INTEGER EXTERNAL ,
  error_check4 POSITION (31:41) DECIMAL EXTERNAL ,
  )

```

Reconciling Payment Information

Once you have successfully completed the set up steps, you can use the AutoClear form to automatically reconcile your payment history with your bank payment clearing information.

When you submit AutoClear, you can choose whether you want to load the reconciliation file, load the file and reconcile payments, or simply reconcile payments.



AutoClear

(Oracle Payables Reference Manual or Oracle Government Payables Reference Manual)

After it loads the payment information from your bank reconciliation file, AutoClear compares the information in the interface tables to the payment information in your Oracle Payables application and updates the status of any payments that match.

Matching Payment Information

AutoClear compares the records loaded into AP_RECON_DETAILS with the payment records in AP_CHECKS_ALL to determine if the records match. AP_CHECKS_ALL contains information about each payment you create in your Oracle Payables application. The following columns must have the same values for a record in AP_RECON_DETAILS to match with a record in AP_CHECKS_ALL:

AP_RECON_DETAILS		AP_CHECKS_ALL	
SERIAL_NUMBER	=	CHECK_NUMBER	
BANK_ACCOUNT_ID	=	BANK_ACCOUNT_ID	
CURRENCY_CODE	=	CURRENCY_CODE	
AMOUNT	=	AMOUNT	

Table 7 – 3 Matching AP_RECON_DETAILS to AP_CHECKS_ALL (Page 1 of 1)

Updating Payment History and Cleared Status

If a record in AP_CHECKS_ALL matches a record in AP_RECON_DETAILS, your Oracle Payables application updates both records.

In AP_RECON_DETAILS, your Oracle Payables application enters RECONCILED in the RECORD_STATUS column to indicate that it successfully reconciled the record with the payment record.

In AP_CHECKS_ALL, your Oracle Payables application clears the payment record by entering:

- RECONCILED BY BANK in STATUS_CODE
- AP_RECON_DETAILS.AMOUNT in CLEARED_AMOUNT
- AP_RECON_DETAILS.CLEARED_DATE in CLEARED_DATE

Identifying Reconciliation Exceptions

If a record in AP_RECON_DETAILS doesn't match with a record in AP_CHECKS_ALL, AutoClear identifies the record as an exception and enters an exception status in the STATUS_CODE column. You can review the status for an exception in the reconciliation reports that your Oracle Payables application automatically prints during AutoClear:

- Payment Reconciliation Exceptions Report

- Reconciliation Summary Report

You can also submit these reports, along with the Payment Reconciliation Detail Report, from the Resolve AutoClear Exceptions form, which you use to manually reconcile exception records.



Resolve AutoClear Exceptions
(*Oracle Payables Reference Manual* or *Oracle Government Payables Reference Manual*)



Payment Reconciliation Detail Report
Payment Reconciliation Exceptions Report
Reconciliation Summary Report
(*Oracle Payables Reference Manual* or *Oracle Government Payables Reference Manual*)

Additional Reconciliation Reports

You can submit the Payment Exceptions Reports at any time to review your exception payments. In addition, your Oracle Payables application provides the AutoClear Code Listing which prints all the record and transaction codes you have defined for your bank accounts.



Run Reports
(*Oracle Payables Reference Manual* or *Oracle Government Payables Reference Manual*)



Payment Exceptions Report
AutoClear Code Listing
(*Oracle Payables Reference Manual* or *Oracle Government Payables Reference Manual*)

Importing Invoice Information into Oracle Payables or Oracle Government Payables

Invoice Import lets you integrate Oracle Payables or Oracle Government Payables with other accounting systems to allow you to import invoices from these systems. You use Invoice Import to create invoices from invoice information that you load into your Oracle Payables application from your other accounting systems. Once you create invoices in your Oracle Payables application, you can pay the invoices and create journal entries for your transactions. This section describes in detail how to import invoices from another accounting system using SQL*Loader and Invoice Import.



Invoice Import
(*Oracle Payables Reference Manual* or *Oracle Government Payables Reference Manual*)



Submit Invoice Import
(*Oracle Payables Reference Manual* or *Oracle Government Payables Reference Manual*)

Basic Integration Needs

Invoice Import provides you with the features you need to satisfy the following basic integration needs:

- Import invoice data quickly and automatically convert the valid data you import into invoices that you can process in your Oracle Payables application
- Import invoice data from a variety of environments, including your own environment and other accounting systems
- Import voucher number information from your feeder systems and assign the information to invoices created by Invoice Import or have your Oracle Payables application automatically assign sequential voucher numbers to Invoice Import invoices
- Import US Standard General Ledger information from your other accounting systems and assign the information to the invoices created in your Oracle Payables application, if you are using Oracle Government Payables and Oracle Government General Ledger

- Import currency information from your other accounting systems to create foreign currency invoices in your Oracle Payables application
- Validate all imported information against defined invoice and vendor controls
- Display the results of your import run. You can identify which invoices have been successfully created, and any errors which may have occurred during the import process.
- Choose to archive your source data each time you run Invoice Import

Major Features

Simple Integration

Invoice Import automatically creates invoices from invoice information you load into your Oracle Payables application. And, Invoice Import automatically creates invoice distribution lines from the line information you load.

Invoice Import Sources

You can define a separate source for each non-Oracle accounting system you use. You then specify a source during Invoice Import to import invoice information that you load from this source.

Voucher Numbers and Sequential Numbering

Invoice Import allows you to manually assign voucher numbers to imported invoices, if you are not using Sequential Numbering. If you are using Sequential Numbering, Invoice Import automatically assigns validated sequential numbers to your invoices during import. Invoice Import also allows you to override the sequential numbering for your imported invoices if you enable the appropriate system option.

USSGL Transaction Codes

Invoice Import allows you to comply with government requirements for the U.S. Standard General Ledger (USSGL). Invoice Import imports transaction code information for your invoices and your Oracle

General Ledger application uses the information to create journal entries for the U.S. Standard General Ledger.

Foreign Currency Invoices

With Invoice Import, you can create invoices for all your foreign currency invoices from your other accounting system. You can record exchange rate and date information, as well as foreign currency amounts, for each invoice.

Complete Validation

Invoice Import ensures that the information you use to create invoices is accurate and valid within your Oracle Payables application. If any of the information is invalid, your Oracle Payables application does not create an invoice from the data and prints a report listing these exceptions.

Invoice Import Reports

Each time you run Invoice Import, your Oracle Payables application prints an Invoice Import Report and an Invoice Import Exception Report.

The Invoice Import Report provides you with detail information for each invoice created by Invoice Import as well as the total number and amounts of the new invoices. The Invoice Import Exception Report informs you of any exceptions that Invoice Import encountered during the import process and gives you an explanation of each exception. You must correct the exception and resubmit Invoice Import to successfully import the invoice.

Audit Information

You can set up Invoice Import to retain imported invoice information after it creates invoices so you have a complete audit trail of the information you transfer from your other accounting systems. You can also have Invoice Import automatically purge imported information once it successfully creates invoices from the information.

Invoice Import Definitions

To understand how Invoice Import works, you should be familiar with the following definitions. For additional definitions, see the Importing Information into Oracle Financials Applications essay in this manual.



Importing Information into Oracle Financials Applications
(See page 1 – 2)

Invoice Import

The Oracle Payables application process that you submit to import invoice information from interface tables into the appropriate AP tables, thereby creating invoices from the imported invoice information. You can use Invoice Import to create invoices from Xpense Xpress, Oracle Web Employees, and Oracle Projects expense reports or from invoice information from another accounting system.

Interface Tables

The tables in your Oracle Payables application to which you load the invoice data in your invoice flat file. You use SQL*Loader to load the invoice information into these tables. The interface tables also store information for your Xpense Xpress, Oracle Web Employees, and Oracle Projects expense reports.

Import Program

The program that you create to compile invoice data from your non-Oracle accounting system(s) and produce the invoice flat file that you load into your Oracle Payables application.

Invoice Flat File

The output from your import program. The file must contain all the necessary data, such as invoice number, invoice amount, and expense accounts, for creating invoices in your Oracle Payables application.

Control File

The SQL*Loader file you create for loading the data in your invoice flat file into the Invoice Import interface tables. The control file serves as a guide for formatting the data in the invoice flat file into the standard format used by the columns in the interface tables.

Importing Invoices into Oracle Payables Applications

Invoice Import depends on invoice data from your accounting system to create corresponding invoices in your Oracle Payables application. You must create an import program that provides this information. Your program should produce a flat file containing the data from your non-Oracle accounting system that Invoice Import needs to create invoices in your Oracle Payables application. You should enlist the aid of your on-site MIS personnel in developing this program or you may wish to use an Oracle consultant.

Understanding the Interface Tables

You use SQL*Loader to load invoice information into the interface tables, AP_EXPENSE_REPORT_HEADERS_ALL and AP_EXPENSE_REPORT_LINES_ALL, for each invoice you want to create. When you initiate importing invoices, Invoice Import validates all required invoice data in these tables. After it validates the information, Invoice Import creates invoices from the information in AP_EXPENSE_REPORT_HEADERS_ALL and invoice distribution lines from the information in AP_EXPENSE_REPORT_LINES_ALL.

The interface tables are organized by columns which your Oracle Payables application uses to categorize and store specific invoice information. For example, invoice source information is stored in the column called SOURCE in AP_EXPENSE_REPORT_HEADERS_ALL.

NOT NULL Columns

You must enter values for all NOT NULL columns in the interface tables to successfully create invoices and invoice distribution lines from your imported invoice data.

NULL Columns

A column in the interface tables that must not have a value in order for Invoice Import to successfully create invoices from the records in the tables.

Required Columns

In addition to the NOT NULL columns, there are columns in the interface tables that require a value for Invoice Import.

Conditionally Required Columns

Your Oracle Payables requires you to enter values in conditionally required columns only if you have entered a value in another column on which the columns are dependent.

Optional Columns

You can use some columns in the interface tables to import additional invoice information for the invoices created by Invoice Import. Invoice Import imports the data that you load into these optional columns, provided that the information passes the validation checks that Invoice Import requires.

Columns Not Used for Invoice Import

Invoice Import does not use certain columns in the interface tables for creating invoices from your other accounting systems. Your Oracle Payables application does not support importing any data from these columns and you should leave these columns null for the records you load into the table.

Table Descriptions

The following section describes the columns in the Invoice Import interface tables and indicates the columns that require values for importing invoices into your Oracle Payables application. For a complete description of the Invoice Import interface tables, please consult the *Oracle Payables Applications Technical Reference Manual*.



Table Definitions
(*Oracle Payables Applications Technical Reference Manual*)

AP_EXPENSE_REPORT_HEADERS_ALL

Column Name	Null	Type	Comments
REPORT_HEADER_ID	NOT NULL	NUMBER	Primary Key
EMPLOYEE_ID		NUMBER	Null
WEEK_END_DATE	NOT NULL	DATE	

Table 7 – 4 AP_EXPENSE_REPORT_HEADERS_ALL (Page 1 of 3)

Column Name	Null	Type	Comments
CREATION_DATE	NOT NULL	DATE	Who column
CREATED_BY	NOT NULL	NUMBER	Who column
LAST_UPDATE_DATE	NOT NULL	DATE	Who column
LAST_UPDATED_BY	NOT NULL	NUMBER	Who column
VOUCHNO	NOT NULL	NUMBER	0
TOTAL	NOT NULL	NUMBER	
VENDOR_ID		NUMBER(15)	Required, Foreign key: PO_VENDORS
VENDOR_SITE_ID		NUMBER(15)	Required, Foreign key: PO_VENDOR_SITES
EXPENSE_CHECK_ ADDRESS_FLAG		VARCHAR2(30)	Not used
REFERENCE_1		NUMBER(15)	Projects reference
REFERENCE_2		VARCHAR2(30)	Projects reference
INVOICE_NUM		VARCHAR2(50)	Required
EXPENSE_REPORT_ID		NUMBER	Not used
ACCTS_PAY_ CODE_COMBINATION_ID		NUMBER(15)	Required
SET_OF_BOOKS_ID		NUMBER(15)	Required
SOURCE		VARCHAR2(25)	Required
PURGEABLE_FLAG		VARCHAR2(1)	Y or N
ACCOUNTING_DATE		DATE	Required
MAXIMUM_ AMOUNT_TO_APPLY		NUMBER	Not used
ADVANCE_INVOICE_ TO_APPLY		NUMBER(15)	Not used
APPLY_ADVANCES_ DEFAULT		VARCHAR2(4)	Not used
EMPLOYEE_CCID		NUMBER	Not used
DESCRIPTION		VARCHAR2(240)	
REJECT_CODE		VARCHAR2(25)	Not used
HOLD_LOOKUP_CODE		VARCHAR2(25)	
ATTRIBUTE_CATEGORY		VARCHAR2(150)	
ATTRIBUTE1 - 15		VARCHAR2(150)	

Table 7 – 4 AP_EXPENSE_REPORT_HEADERS_ALL (Page 2 of 3)

Column Name	Null	Type	Comments
DEFAULT_CURRENCY_CODE	NOT NULL	VARCHAR2(15)	
DEFAULT_EXCHANGE_RATE_TYPE		VARCHAR2(30)	Conditionally required
DEFAULT_EXCHANGE_RATE		NUMBER	Conditionally required
DEFAULT_EXCHANGE_DATE		DATE	Conditionally required
LAST_UPDATE_LOGIN		NUMBER(15)	Not used, Who column
VOUCHER_NUM		VARCHAR2(50)	
USSGL_TRANSACTION_CODE		VARCHAR2(30)	
USSGL_TRX_CODE_CONTEXT		VARCHAR2(30)	Not used
DOC_CATEGORY_CODE		VARCHAR2(30)	
WORKFLOW_AP-PROVED_FLAG		VARCHAR2(1)	
FLEX_CONCATENATED		VARCHAR2(240)	
ORG_ID		NUMBER	Multiple Organization Support

Table 7 – 4 AP_EXPENSE_REPORT_HEADERS_ALL (Page 3 of 3)

NOT NULL Columns **CREATED_BY**

(AP_EXPENSE_REPORT_HEADERS_ALL)

Enter the ID of the person who loads this record into the table. Invoice Import transfers this ID to the AP_INVOICES_ALL and AP_PAYMENT_SCHEDULES_ALL tables during import so that the creator of the record becomes the invoice and payment schedule creator. Similarly, if you enter a HOLD_LOOKUP_CODE for the record, Invoice Import transfers this ID to the AP_HOLDS_ALL tables so that the record creator of the record becomes the hold creator.

Validation: None

Destination: AP_INVOICES_ALL.CREATED_BY,
AP_PAYMENT_SCHEDULES_ALL.CREATED_BY,
AP_HOLDS_ALL.CREATED_BY

CREATION_DATE

Enter the date on which you load this record into the table. Your Oracle Payables application uses this date for reference and audit purposes. When Invoice Import creates an invoice from this record, it does not use this date as the creation date for the invoice; it uses the system date at the time you submit Invoice Import.

Your Oracle Payables application also uses this date to determine the records to purge if you choose to purge records during import.

Validation: Must be in valid date format.

Destination: None

DEFAULT_CURRENCY_CODE

Enter the currency code for your invoice. Enter a currency code that is different from your functional currency code to create foreign currency invoices.

If you enter a foreign currency code and you have enabled your Require Exchange Rate Entry system option, you must enter a value for DEFAULT_EXCHANGE_RATE. You should also enter values for:

- DEFAULT_EXCHANGE_DATE
- DEFAULT_EXCHANGE_RATE_TYPE

Validation: If you enter a code for a foreign currency, you must enter a valid currency code. You can obtain a list of valid currency codes from FND_CURRENCIES.CURRENCY_CODE.

Destination: AP_INVOICES_ALL.INVOICE_CURRENCY_CODE

LAST_UPDATED_BY

Enter the ID of the person who last updated this record (usually the same value as CREATED_BY).

Validation: None

Destination: AP_INVOICES_ALL.LAST_UPDATED_BY,
AP_PAYMENT_SCHEDULES_ALL.LAST_UPDATED_BY,
AP_HOLDS_ALL.LAST_UPDATED_BY

LAST_UPDATE_DATE

Enter the last update date for this record (usually the same date as CREATION_DATE). Your Oracle Payables application uses this date for reference and audit purposes only. When Invoice Import creates an invoice from a record in this table, it does not use this date as the last update date for the invoice; it uses the system date at the time you submit Invoice Import.

Validation: Must be in valid date format.

Destination: None

REPORT_HEADER_ID

Enter the ID number for each record you load to this table. Invoice Import does not use this number as the invoice ID; instead, it uses the number to uniquely identify the records in the table.

Validation: Must be unique for all records in the table.

Destination: None



Suggestion: We recommend that you use the Sequence Generator, AP_EXPENSE_REPORT_HEADERS_S, to obtain a value for this column.

TOTAL

Enter the total amount of your invoice. If the amount you enter is greater than or equal to zero, Invoice Import creates an invoice with the invoice type Standard. If the amount you enter is less than zero, Invoice Import creates a Credit type invoice.

If you wish to import a foreign currency invoice, you should enter this amount in your foreign currency. Your Oracle Payables application automatically converts the foreign amount to your functional amount, based on the exchange rate you enter for the invoice.

Validation: None

Destination: AP_INVOICES_ALL.INVOICE_AMOUNT

VOUCHNO

Enter 0 (zero) for all the records you load to this table. A value of zero indicates that an invoice has not been created from the record. Invoice Import only creates invoices for records with a zero value in this column. When Invoice Import successfully creates an invoice, it

automatically updates this value to be the INVOICE_ID for the invoice created from this record.



Attention: This column does not hold the value for the voucher number for an invoice. You enter the value for a voucher number you want to assign to an imported invoice in the VOUCHER_NUM column.

Validation: None

Destination: None

WEEK_END_DATE

Enter the date that you want to assign to your invoice as the invoice date. This date also serves as the basis for the GL Date on the invoice distribution lines created for this invoice, unless you specify a GL Date when you initiate Invoice Import or enter a date in the ACCOUNTING_DATE column.

Validation: Must be in valid date format. If your Oracle Payables application uses this date as the GL_DATE for the distribution lines created for this invoice, the date you enter must be in an open or future period.

Destination: AP_INVOICES_ALL.INVOICE_DATE

NULL Columns

(AP_EXPENSE_REPORT_HEADERS_ALL)

EMPLOYEE_ID

Do not enter a value in this column. A value in this column indicates that the record is used for an employee expense report.

Validation: Must be NULL

Destination: None

Required Columns

(AP_EXPENSE_REPORT_HEADERS_ALL)

ACCOUNTING_DATE

Enter the accounting date for your invoice. Invoice Import assigns this accounting date to all the invoice distribution lines that it creates for this invoice, unless you specify a GL Date when you submit Invoice Import.

Validation: Must be in valid date format and in a future or open period.

Destination: AP_INVOICE_DISTRIBUTIONS_ALL.
ACCOUNTING_DATE

ACCTS_PAY_CODE_COMBINATION_ID

Enter the Flexfield ID number for the accounts payable liability account you want to assign to your invoice.

- Validation:** The ID you enter must be for a valid Flexfield combination. You can obtain a list of valid accounts from GL_CODE_COMBINATIONS.CODE_COMBINATION_ID.
- Destination:** AP_INVOICES_ALL.ACCTS_PAY_CODE_COMBINATION_ID

INVOICE_NUM

Enter the invoice number that you want to assign to the invoice created in your Oracle Payables application from this record. If you choose to leave this column blank, your Oracle Payables application uses the WEEK_END_DATE as the invoice number.

- Validation:** You must assign a unique number for the supplier to the invoice. If you assign a duplicate number for the supplier, Invoice Import does not create an invoice from this record and lists this record on the Invoice Import Exceptions Report.
- Destination:** AP_INVOICES_ALL.INVOICE_NUM

SET_OF_BOOKS_ID

Enter the appropriate set of books ID for your invoice.

- Validation:** The ID you enter must be for the set of books you define in the Define Financials Options form.
- Destination:** AP_INVOICES_ALL.SET_OF_BOOKS_ID

SOURCE

Enter the name of the Source code you have defined for invoices that you import from your other accounting system. The source of an record determines the type of invoice created from the record. For each record with the Source code you define for your other accounting system, Invoice Import creates an invoice with either the type STANDARD or CREDIT. If the amount for the record is positive or zero, Invoice Import creates a standard invoice. If the amount is negative, Invoice Import creates a credit invoice.

The Source code name also determines which records will be selected for import. You specify a Source code name when you initiate Invoice Import.

Validation: You must enter the name exactly as you have defined the code in the Define Payables QuickCodes form. If the source you enter does not match the name as you defined it in the Define Payables QuickCodes form, when you submit Invoice Import for this source, Invoice Import will not create an invoice from the record.

Destination: AP_INVOICES_ALL.SOURCE



Define Payables QuickCodes
Submit Invoice Import
(*Oracle Payables Reference Manual* or *Oracle Government Payables Reference Manual*)

VENDOR_ID

Enter the vendor ID for the vendor you wish to assign to the invoice you are importing. If you have not yet entered the vendor for an invoice you want to import, you must enter the vendor in your Oracle Payables application before you can import an invoice for the vendor.

Validation: The ID you enter must be for an existing, valid vendor. You can obtain a list of valid values from PO_VENDORS.VENDOR_ID.

Destination: AP_INVOICES_ALL.VENDOR_ID



Invoice Import
(*Oracle Payables Reference Manual* or *Oracle Government Payables Reference Manual*)

VENDOR_SITE_ID

Enter the vendor site ID for the vendor site you wish to assign to the invoice you are importing.

Validation: The ID you enter must be for an existing, valid vendor site for the vendor you specify (VENDOR_ID). You can obtain a list of valid values for a vendor from PO_VENDOR_SITES.VENDOR_SITE_ID.

Destination: AP_INVOICES_ALL.VENDOR_SITE_ID

Conditionally Required Columns

(AP_EXPENSE_REPORT_HEADERS_ALL)

DEFAULT_EXCHANGE_DATE

You should enter a value in this column if you enter a foreign currency code in the DEFAULT_CURRENCY_CODE column. The date you enter determines the exchange rate for a foreign currency invoice with a Spot, Corporate, Other, or user-defined rate type.

Validation: Must be in valid date format.

Destination: AP_INVOICES_ALL.EXCHANGE_DATE

DEFAULT_EXCHANGE_RATE

This column is required if you enter a foreign currency code in the DEFAULT_CURRENCY_CODE column and your Require Rate Entry system option is enabled.

If you enter a value in this field, you should enter USER in the DEFAULT_EXCHANGE_RATE_TYPE column. If you enter a type other than USER in DEFAULT_EXCHANGE_RATE_TYPE, you should leave this column NULL.

Validation: You must enter a value in this column if you enter a foreign currency code for this record and your Require Rate Entry system option is enabled. In addition, if you enter User in the DEFAULT_EXCHANGE_RATE_TYPE column, you should enter a rate in this field.

Destination: AP_INVOICES_ALL.EXCHANGE_RATE

DEFAULT_EXCHANGE_RATE_TYPE

You should enter a value in this column if you enter a foreign currency code in the DEFAULT_CURRENCY_CODE column. You can enter a predefined or user-defined exchange rate type. The exchange rate type determines which exchange rate you use. If you enter a type other than USER, you should leave the DEFAULT_EXCHANGE_RATE column null.

Validation: You must enter a predefined (SPOT, CORPORATE, USER, or OTHER) or user-defined rate type.

Destination: AP_INVOICES_ALL.EXCHANGE_RATE_TYPE



Define Daily Conversion Rate Types

(Oracle Payables Reference Manual or Oracle Government
Payables Reference Manual)

DOC_CATEGORY_CODE

If you are using Sequential Numbering, Invoice Import uses this column to assign a document category to each invoice it creates. The document category for an invoice determines whether your Oracle Payables application uses an automatic sequential number or allows you to enter a manual sequential number for an invoice.

For invoices with positive invoice amounts, Invoice Import assigns the document category for standard invoices. For negative amount invoices, Invoice Import assigns the credit invoice document category.

If you enable the Document Category Override system option, you can enter the document category you want Invoice Import to assign, in place of the standard or credit document category, to the invoice created from this record.

Validation: Do not enter a value in this column if you are using Sequential Numbering and you do not enable the Document Category Override system option for your invoices. If you enter a value into this column under these circumstances, Invoice Import identifies the record as an exception and does not create an invoice from the record.

In addition, if you are not using Sequential Numbering, you should leave this column blank.

Destination: AP_INVOICES_ALL.DOC_CATEGORY_CODE

ORG_ID

If you use the Oracle Applications multiple organizations feature, enter the operating unit's org_id. If you do not use the Oracle Applications multiple organizations feature, leave this column null. To insert rows into open interface tables, use the multiple organization AP_EXPENSE_REPORT_HEADERS_ALL table and explicitly populate the org_id column. You can determine an operating unit's org_id by querying the HR_OPERATING_UNITS view:

```
SELECT name, organization_id from  
HR_OPERATING_UNITS;
```

When you import rows *from* open interface tables, only rows from your current operating unit are processed.

Optional Columns

(AP_EXPENSE_REPORT_HEADERS_ALL)

ATTRIBUTE [1–15]

Enter Descriptive Flexfield information that you want to import for an invoice. The structure of the information you enter in these columns (datatypes, value sets) should match the structure of the Descriptive Flexfield segments you have defined for your invoices or you will experience validation problems when you try to access the information in the invoice forms.

Validation: None

Destination: AP_INVOICES_ALL.ATTRIBUTE[1–15]

ATTRIBUTE_CATEGORY

Enter the Descriptive Flexfield category for the Descriptive Flexfield information you want to import.

Validation: None

Destination: AP_INVOICES_ALL.ATTRIBUTE_CATEGORY

DESCRIPTION

Enter the description that you want to assign to the invoice created from this record.

Validation: None

Destination: AP_INVOICES_ALL.DESCRPTION

HOLD_LOOKUP_CODE

Enter the name of a hold reason that you want to assign to the invoice created from this record. You can only assign one hold to the invoice created from this record. Once Invoice Import create the invoices, you can enter additional holds in the Enter Invoices form.

Invoice Import transfers this code to AP_HOLDS_ALL, along with INVOICE_ID for the invoice created from this record, to create a hold on the invoice. You must release the hold before you can pay the invoice.

Validation: The name you enter must be for an active Invoice hold reason. You can obtain a listing of valid Invoice hold listings from
AP_HOLD_CODES.HOLD_LOOKUP_CODE
(HOLD_TYPE = INVOICE HOLD REASON)

Destination: AP_HOLDS_ALL.HOLD_LOOKUP_CODE

Projects Columns

Enter projects information from Oracle Projects in the following columns if you want to associate the invoice (created from this record) with a project in Oracle Projects.

- REFERENCE_1
- REFERENCE_2

Your Oracle Payables application transfers the information into the AP_INVOICES_ALL table.

Validation: None

Destination: AP_INVOICES_ALL.[Projects Columns]



Integrating with Oracle Purchasing and Oracle Payables
(*Oracle Projects Reference Manual*)

PURGEABLE_FLAG

Enter a value (Y or N) to indicate whether this record and records in AP_EXPENSE_REPORT_LINES_ALL with the same REPORT_HEADER_ID are purgeable during Invoice Import. If you set this column to Y or leave it null, you can purge these records when you submit Invoice Import.

Your Oracle Payables application purges records during import if the following criteria are met:

- Invoice Import successfully created an invoice from the record (VOUCHNO <> 0)
- PURGEABLE_FLAG for the record set to Y or NULL
- Either the source you specify in the Submit Invoice Import form is Payables Expense Report or Oracle Projects, and the record SOURCE is the same; or the source you specify in the Submit Invoice Import form is a user-defined QuickCode (regardless of the record SOURCE).
- CREATION_DATE falls before the Purge Date you specify in the Submit Invoice Import form.

Validation: Must be Y or N

Destination: None



Submit Invoice Import
(*Oracle Payables Reference Manual* or *Oracle Government Payables Reference Manual*)

USSGL_TRANSACTION_CODE

Enter the default transaction code you want to assign to an invoice.

If you use Oracle Government General Ledger with your Oracle Government Payables, you may be required to maintain US Standard General Ledger information for your invoice transactions. Invoice Import imports any USSGL information that you load into the interface tables and assigns the information to the invoices it creates.

Validation: None

Destination: AP_INVOICES_ALL.USSGL_TRANSACTION_CODE

VOUCHER_NUM

Enter a value for the voucher number you want to apply to the invoice created from this record. If you use manual Sequential Numbering, this number should be unique.

Validation: If you use automatic Sequential Numbering, Payables ignores any value you load into this column and instead selects the next available number for the numbering sequence assigned to the document category for invoices imported from other accounting systems. If you use manual sequential numbering, Payables validates for uniqueness but does not require the next available number. If you do not enable the Sequential Numbering profile option, the system does not validate the number.

Destination: AP_INVOICES_ALL.VOUCHER_NUM

AP_EXPENSE_REPORT_LINES_ALL

Column Name	Null	Type	Comments
REPORT_HEADER_ID	NOT NULL	NUMBER	
LAST_UPDATE_DATE	NOT NULL	DATE	Who column
LAST_UPDATED_BY	NOT NULL	NUMBER	Who column
CODE_COMBINATION_ID	NOT NULL	NUMBER	
ITEM_DESCRIPTION	NOT NULL	VARCHAR2(240)	

Table 7 – 5 AP_EXPENSE_REPORT_LINES_ALL (Page 1 of 3)

Column Name	Null	Type	Comments
SET_OF_BOOKS_ID	NOT NULL	NUMBER(15)	
AMOUNT		NUMBER	
ATTRIBUTE_CATEGORY		VARCHAR2(150)	
ATTRIBUTE1 - 15		VARCHAR2(150)	
CURRENCY_CODE	NOT NULL	VARCHAR2(15)	
EXCHANGE_RATE_TYPE		VARCHAR2(30)	Not used
EXCHANGE_RATE		NUMBER	Not used
EXCHANGE_DATE		DATE	Not used
VAT_CODE		VARCHAR2(15)	
LINE_TYPE_LOOKUP_CODE	NOT NULL	VARCHAR2(25)	
LAST_UPDATE_LOGIN		NUMBER(15)	Not used, Who column
CREATION_DATE		DATE	Required, Who column
REFERENCE_1		NUMBER(15)	Projects reference
REFERENCE_2		VARCHAR2(30)	Projects reference
CREATED_BY		NUMBER(15)	Required, Who column
STAT_AMOUNT		NUMBER	
USSGL_TRANSACTION_CODE		VARCHAR2(30)	
USSGL_TRX_CODE_CONTEXT		VARCHAR2(30)	Not used
PROJECT_ACCOUNTING_CONTEXT		VARCHAR2(30)	Projects
PROJECT_ID		NUMBER(15)	Projects
TASK_ID		NUMBER(15)	Projects
EXPENDITURE_ORGANIZATION_ID		NUMBER(15)	Projects
EXPENDITURE_TYPE		VARCHAR2(30)	Projects
EXPENDITURE_ITEM_DATE		DATE	Projects
PA_QUANTITY		NUMBER	Projects
DISTRIBUTION_LINE_NUMBER		NUMBER(15)	Null

Table 7 – 5 AP_EXPENSE_REPORT_LINES_ALL (Page 2 of 3)

Column Name	Null	Type	Comments
ADJUSTMENT_REASON		VARCHAR2(240)	Web Employees
JUSTIFICATION_REQUIRED_FLAG		VARCHAR2(1)	Web Employees
RECEIPT_REQUIRED_FLAG		VARCHAR2(1)	Web Employees
RECEIPT_VERIFIED_FLAG		VARCHAR2(1)	Web Employees
RECEIPT_MISSING_FLAG		VARCHAR2(1)	Web Employees
JUSTIFICATION		VARCHAR2(240)	Web Employees
EXPENSE_GROUP		VARCHAR2(80)	Web Employees
START_EXPENSE_DATE		DATE	Web Employees
END_EXPENSE_DATE		DATE	Web Employees
RECEIPT_CURRENCY_CODE		VARCHAR2(15)	Web Employees
RECEIPT_CONVERSION_RATE		NUMBER	Web Employees
DAILY_AMOUNT		NUMBER	Web Employees
RECEIPT_CURRENCY_AMOUNT		NUMBER	Web Employees
WEB_PARAMETER_ID		NUMBER	Web Employees
AMOUNT_IN-CLUDES_TAX_FLAG		VARCHAR2(1)	
ORG_ID		NUMBER	Multiple Organization Support

Table 7 – 5 AP_EXPENSE_REPORT_LINES_ALL (Page 3 of 3)

NOT NULL Columns

(AP_EXPENSE_REPORT_LINES_ALL)

CODE_COMBINATION_ID

Enter the ID number for the account to which you want to charge this invoice distribution line.

Validation: The ID you enter must be for a valid Accounting Flexfield combination. You can obtain a list of valid accounts from GL_CODE_COMBINATIONS.CODE_COMBINATION_ID.

Destination: AP_INVOICE_DISTRIBUTIONS_ALL.DISTCODE_COMBINATION_ID

CURRENCY_CODE

Enter the currency code for the invoice distribution line.

Validation: You must enter the same currency_code as you enter for DEFAULT_CURRENCY_CODE in AP_EXPENSE_REPORT_HEADERS_ALL.

Destination: AP_EXPENSE_REPORT_LINES.CURRENCY_CODE

ITEM_DESCRIPTION

Enter a description that you want to assign to the invoice distribution created from this record.

Validation: None

Destination: AP_INVOICE_DISTRIBUTIONS_ALL.DESCRPTION

LAST_UPDATED_BY

Enter the ID of the person who last updated this record (usually the same value as CREATED_BY).

Validation: None

Destination: AP_INVOICE_DISTRIBUTIONS_ALL.LAST_UPDATED_BY

LAST_UPDATE_DATE

Enter the last update date for this record (usually the same date as CREATION_DATE). Your Oracle Payables application uses this date for reference and audit purposes only. When Invoice Import creates an invoice distribution from a record in this table, it does not use this date as the last update date for the invoice; it uses the system date at the time you submit Invoice Import.

Validation: None

Destination: None

LINE_TYPE_LOOKUP_CODE

Enter the lookup code for the type of invoice distribution line that you want Invoice Import to create from this record.

Validation: The code you enter must be ITEM, TAX, MISC, or FREIGHT. These lookup codes are stored in the AP_LOOKUP_CODES table.

Destination: AP_INVOICE_DISTRIBUTIONS_ALL.LINE_TYPE
_LOOKUP_CODE

REPORT_HEADER_ID

Enter the unique ID number you entered for each record in the AP_EXPENSE_REPORT_HEADERS_ALL table. This number identifies the invoice for this distribution line.

Validation: The ID you enter must be an ID for an existing record in
AP_EXPENSE_REPORT_HEADERS_ALL.

Destination: None

SET_OF_BOOKS_ID

Enter the appropriate set of books ID for your invoice distribution lines.

Validation: The ID you enter must be for the set of books you define in the Define Financials Options form.

Destination: AP_INVOICE_DISTRIBUTIONS_ALL.SET_OF_BOOKS_ID

NULL Columns

(AP_EXPENSE_REPORT_LINES_ALL)

DISTRIBUTION_LINE_NUMBER

Do not enter a value in this column. Your Oracle Payables application uses this column to assign the DISTRIBUTION_LINE_NUMBER for the invoice distribution line created from this record. Before Invoice Import creates an invoice distribution line from this record, your Oracle Payables application updates this column with a value and then copies the value to AP_INVOICE_DISTRIBUTIONS_ALL for the invoice distribution line.



Attention: Your Oracle Payables application does not use this number to link the invoice distribution line created from a record in AP_EXPENSE_REPORT_LINES_ALL to the original record. Once Invoice Import successfully creates an invoice distribution line, you can update the distribution line number for the line.

Validation: None

Destination: AP_INVOICE_DISTRIBUTIONS_ALL.
DISTRIBUTION_LINE_NUMBER.

Required Columns

(AP_EXPENSE_REPORT_LINES_ALL)

CREATED_BY

Enter the ID of the person who loads this record into the table. Invoice Import transfers this ID to the AP_INVOICE_DISTRIBUTIONS_ALL table during import so that the creator of the invoice distribution record becomes the invoice distribution line creator.

Validation: None

Destination: AP_INVOICE_DISTRIBUTIONS_ALL.CREATED_BY

CREATION_DATE

Enter the date on which you load this record into the table. Your Oracle Payables application uses this date for reference and audit purposes. When Invoice Import creates an invoice distribution line from this record, it does not use this date as the creation date for the line; it uses the system date at the time you submit Invoice Import.

Validation: Must be a valid date format

Destination: None

Conditionally Required Columns

(AP_EXPENSE_REPORT_LINES_ALL)

ORG_ID

If you use the Oracle Applications multiple organizations feature, enter the operating unit's org_id. If you do not use the Oracle Applications multiple organizations feature, leave this column null. To insert rows into open interface tables, use the multiple organization AP_EXPENSE_REPORT_LINES_ALL table and explicitly populate the org_id column. You can determine an operating unit's org_id by querying the HR_OPERATING_UNITS view:

```
SELECT name, organization_id from  
HR_OPERATING_UNITS;
```

When you import rows *from* open interface tables, only rows from your current operating unit are processed.

Optional Columns

(AP_EXPENSE_REPORT_LINES_ALL)

AMOUNT

Enter an amount for the invoice distribution line created from this record. Your Oracle Payables application charges the amount you enter to the Accounting Flexfield you specify for this record. You can enter any amount for an invoice distribution line; however, if the total amount of all the invoice distribution lines does not equal the amount

of the invoice, your Payables application places the invoice on hold during AutoApproval.

If you do not enter an amount for a record in this table, Invoice Import creates an invoice distribution line for the record and assigns the Accounting Flexfield you specify to this line, but does not record an amount for the line.

Validation: None

Destination: AP_INVOICE_DISTRIBUTIONS_ALL.AMOUNT

ATTRIBUTE [1–15]

Enter Descriptive Flexfield information that you want to import for an invoice distribution line. The structure of the information you enter in these columns (datatypes, value sets) should match the structure of the Descriptive Flexfield segments you have defined for your invoices or you will experience validation problems when you try to access the information in the invoice forms.

Validation: None

Destination: AP_INVOICE_DISTRIBUTIONS_ALL.ATTRIBUTE
[1–15]

ATTRIBUTE_CATEGORY

Enter the Descriptive Flexfield category for the Descriptive Flexfield information you want to import for an invoice distribution line.

Validation: None

Destination: AP_INVOICE_DISTRIBUTIONS_ALL.ATTRIBUTE
_CATEGORY

Projects Columns

Enter project information from Oracle Projects in the following columns if you want to associate the invoice distribution line (created from this record) with a project in Oracle Projects.

- DISTRIBUTION_LINE_NUMBER
- EXPENDITURE_ITEM_DATE
- EXPENDITURE_ORGANIZATION_ID
- EXPENDITURE_TYPE
- PA_QUANTITY

- PROJECT_ACCOUNTING_CONTEXT
- PROJECT_ID
- TASK_ID
- REFERENCE_1
- REFERENCE_2

Validation: None

Destination: AP_INVOICE_DISTRIBUTIONS_ALL.[Pr ojects
Columns]



Integrating with Oracle Purchasing and Oracle Payables
(*Oracle Projects Reference Manual*)

STAT_AMOUNT

Enter the quantity that you want to assign to the statistical unit for an invoice distribution line. When you post the invoice distribution line to your Oracle General Ledger application, your Oracle Payables application transfers this quantity along with the invoice distribution line.

Validation: The account segment for the invoice distribution line Accounting Flexfield must be defined in your Oracle General Ledger to support statistical units of measure.

Destination: AP_INVOICE_DISTRIBUTIONS_ALL.ST AT_
AMOUNT

USSGL_TRANSACTION_CODE

Enter the transaction code you want to assign to an invoice distribution line.

If you use Oracle Government General Ledger with your Oracle Government Payables, you may be required to maintain US Standard General Ledger information for all your invoice transactions. Invoice Import imports any USSGL information that you enter into the interface tables and assigns the information to the invoices it creates. When you post your invoices to Oracle Government General Ledger from Oracle Government Payables, Oracle Government Payables transfers the USSGL information associated with your invoice distribution lines.

Validation: None

Destination: AP_INVOICE_DISTRIBUTIONS.USSGL_TRANSACTION_CODE.

VAT_CODE

Enter the tax name you want to assign to the invoice distribution line created from this record.

Validation: The tax name you enter must be a valid tax name with a type of SALES, USE or user-defined. If you enter an invalid tax name in this column, Invoice Import identifies the record as an exception and does not create an invoice distribution line.

Destination: AP_INVOICE_DISTRIBUTIONS_ALL.VAT_CODE

Oracle Web Employees Columns

The following columns have been added to support integration with Oracle Web Employees.

- ADJUSTMENT_REASON
- DAILY_AMOUNT
- END_EXPENSE_DATE
- EXPENSE_GROUP
- JUSTIFICATION
- JUSTIFICATION_REQUIRED_FLAG
- RECEIPT_CONVERSION_RATE
- RECEIPT_CURRENCY_AMOUNT
- RECEIPT_CURRENCY_CODE
- RECEIPT_MISSING_FLAG
- RECEIPT_MISSING_FLAG
- RECEIPT_REQUIRED_FLAG
- RECEIPT_VERIFIED_FLAG
- START_EXPENSE_DATE
- WEB_PARAMETER_ID

Validation: None

Destination: In this release, Invoice Import does not populate AP_INVOICE_DISTRIBUTIONS_ALL with these

columns. These columns are used only by Oracle Web Employees.

AMOUNT_INCLUDES_TAX_FLAG

Enter a value (Y or N) to indicate whether the amount in the AP_EXPENSE_REPORT_LINES_ALL.AMOUNT column includes tax.

Validation: None.

Destination: AP_INVOICE_DISTRIBUTIONS_ALL.AMOUNT_INCLUDES_TAX_FLAG

Setting Up Invoice Import

The following list summarizes the procedure for setting up Invoice Import to import invoices from your other accounting systems. The rest of the section explains each step in detail.

Setup Checklist

Checklist

- ☐ 1. Define an Invoice Import Source
- ☐ 2. Create a Program for Generating an Invoice Flat File
- ☐ 3. Create SQL*Loader Control File

Setup Steps

Complete the following steps to set up Invoice Import in your Oracle Payables application:

Step 1 **Define an Invoice Import Source**

Use the Define Payables QuickCodes form to define a QuickCode for the source of your Invoice Import. Use any name for the source, except Payables Expense Reports and Oracle Projects. Payables Expense Reports and Oracle Projects are predefined sources that you use to create invoices from expense reports entered in Xpense Xpress, Oracle

Web Employees, or Oracle Projects. Enter Source as your QuickCode Type for the code you define.



Define Payables QuickCodes
(*Oracle Payables Reference Manual* or *Oracle Government Payables Reference Manual*)

You use SQL*Loader to enter this Source code in the SOURCE column of the AP_EXPENSE_REPORT_HEADERS_ALL table. You then select the code in the Submit Invoice Import form when you want to import invoices from this source.

Step 2 Create a Program for Generating an Invoice Flat File

To load invoice information into your Oracle Payables application via SQL*Loader, first create a program that produces a flat file containing the invoice details you want to bring into your Oracle Payables application.

For each invoice, the program should include values in the flat file to populate the following foreign key columns in AP_EXPENSE_REPORT_HEADERS_ALL:

- VENDOR_ID
- VENDOR_SITE_ID
- ACCTS_PAY_CODE_COMBINATION_ID

For each invoice distribution line, include the value to populate the Accounting Flexfield column in AP_EXPENSE_REPORT_LINES_ALL:

- CODE_COMBINATION_ID

In addition, if you are importing foreign currency invoices, include information to populate the following columns in AP_EXPENSE_REPORT_HEADERS_ALL:

- DEFAULT_EXCHANGE_RATE_TYPE
- DEFAULT_EXCHANGE_RATE
- DEFAULT_EXCHANGE_DATE

If this exchange rate information is the same for all invoices and invoice distribution lines you import, you can include this information in your SQL*Loader control file instead.

Finally, ensure that your invoice flat file has the appropriate information to populate AP_EXPENSE_REPORT_HEADERS_ALL and AP_EXPENSE_REPORT_LINES_ALL as indicated in the preceding

table descriptions. If a value is not required for a column, you may leave the column empty.

Step 3 **Create SQL*Loader Control File**

Read the SQL*Loader documentation to learn how to write a SQL*Loader control file. The file you write will vary greatly depending on the nature and format of the invoice flat file you use. Your control file must populate AP_EXPENSE_REPORT_HEADERS_ALL and AP_EXPENSE_REPORT_LINES_ALL as indicated in the previous table descriptions.



Additional Information: SQL*Loader *(ORACLE7 Server Utilities Guide)*

You should code your control file to populate the SOURCE column in AP_EXPENSE_REPORT_HEADERS_ALL with the Source code exactly as you defined it in the Define Payables QuickCodes form.



Define Payables QuickCodes *(Oracle Payables Reference Manual or Oracle Government Payables Reference Manual)*

In addition, EMPLOYEE_ID must be NULL and VOUCHNO must be zero (0) for each invoice you want to import. Your Oracle Payables application only imports invoices that have a value of 0 in the VOUCHNO column.

If all your invoices are in the same currency, you can hard code the value for DEFAULT_CURRENCY_CODE in AP_EXPENSE_REPORT_HEADERS_ALL and CURRENCY_CODE in AP_EXPENSE_REPORT_LINES_ALL in your control file.

You may also want to hard code a value for PURGEABLE_FLAG in AP_EXPENSE_REPORT_HEADERS_ALL to indicate whether the records are purgeable. When you submit Invoice Import, you can choose to delete all purgeable records from the interface tables. A record is purgeable if Invoice Import has successfully created an invoice from the record and the PURGEABLE_FLAG is set to Y or NULL. Consider that you may not want to make these interim records purgeable because they provide an audit trail from your newly created invoices back to the original invoices in your other accounting system.

After Setting Up

Once you have successfully completed the set up steps, you can load your invoice information from your flat file into the interface tables and use Invoice Import to create invoices from the invoice information.



Invoice Import
(*Oracle Payables Reference Manual* or *Oracle Government Payables Reference Manual*)

Integrating Purchase Order Information with Oracle Payables or Oracle Government Payables

Oracle Payables and Oracle Government Payables provide sophisticated matching features to ensure that you only pay for goods and services you ordered, received and accepted. Your Oracle Payables application is tightly integrated with your Oracle Purchasing application to facilitate paperless matching, but you can take advantage of matching even if you are using another purchasing system with your Oracle Payables application.

This document explains the data necessary to integrate your Oracle Payables application with a non-Oracle purchasing system and describes how your Oracle Payables application uses the data to perform two-way, three-way and four-way matching.



Attention: We only include descriptions of the **required** tables and columns for matching to a non-Oracle purchasing system. For complete details on all the entities and relationships that your Oracle Payables application uses for matching, please consult the *Oracle Payables Applications Technical Reference Manual*.



Table Definitions
(*Oracle Payables Applications Technical Reference Manual*)

Understanding the Matching Tables

Your Oracle Payables application uses several of your Oracle Purchasing application tables for matching. To implement matching in Oracle Payables or Oracle Government Payables, you need to load these tables with the data from your non-Oracle purchasing application.

- PO_HEADERS_ALL
- PO_LINES_ALL
- PO_LINE_LOCATIONS_ALL
- PO_DISTRIBUTIONS_ALL
- PO_DISTRIBUTIONS_AP_V (view of PO_DISTRIBUTIONS)
- PO_RELEASES_ALL (Blanket Purchase Orders)
- PO_LOOKUP_CODES

AutoInstall automatically installs these and other necessary Oracle Purchasing application tables when you install your Oracle Payables application.

PO_HEADERS_ALL

Each record in this table represents a purchase order, which is an order for goods or services from a single vendor. Each purchase order may have multiple lines (PO_LINES_ALL).

In addition, each blanket purchase order may have multiple blanket releases (PO_RELEASES_ALL), which release an amount from the blanket.

PO_LINES_ALL

Each record in this table represents a purchase order line, which identifies the items and unit price for the goods ordered on a purchase order. Each purchase order line may have multiple shipments (PO_LINE_LOCATIONS_ALL).

PO_LINE_LOCATIONS_ALL

Each record in this table represents a purchase order shipment, which identifies the quantity of an item shipped to a buyer location by the vendor. Each purchase order shipment may have multiple accounting distributions (PO_DISTRIBUTIONS_ALL).

PO_DISTRIBUTIONS_ALL/PO_DISTRIBUTIONS_AP_V

Each record in this table/view represents a purchase order distribution, which identifies the account charged for the items on a purchase order shipment.

PO_RELEASES_ALL

Each record in this table represents a blanket release for a purchase order. A blanket release may create multiple shipments.

AP_INVOICES_ALL/AP_INVOICE_DISTRIBUTIONS_ALL

Each purchase order shipment can be matched to multiple invoices (AP_INVOICES_ALL), and a single invoice may be matched to multiple purchase order shipments.

When you match an invoice to a purchase order shipment, your Oracle Payables application creates an invoice distribution line (AP_INVOICE_DISTRIBUTIONS_ALL) from each purchase order distribution on the shipment. When you match an invoice to a single purchase order distribution, your Oracle Payables application creates a single invoice distribution line from the purchase order distribution.

Table Descriptions

The following section describes the tables and the columns that your Oracle Payables application supports for matching to purchase orders from your non-Oracle purchase system. We describe how the columns are used and, if a column is required, the values you must load to successfully perform matching. For a complete description of the tables, please consult the *Oracle Payables Applications Technical Reference Manual*.



Attention: You must populate all NOT NULL columns in the purchasing tables.

PO_HEADERS_ALL

Column Name	Null	Type	Comments
PO_HEADER_ID	NOT NULL	NUMBER	Primary key
AGENT_ID	NOT NULL	NUMBER	Foreign key: HR_EMPLOYEES
TYPE_LOOKUP_CODE	NOT NULL	VARCHAR2(25)	Foreign Key: PO_LOOKUP_ CODES
LAST_UPDATE_DATE	NOT NULL	DATE	
LAST_UPDATED_BY	NOT NULL	NUMBER	
SEGMENT1	NOT NULL	VARCHAR2(20)	PO number
SUMMARY_FLAG	NOT NULL	VARCHAR2(1)	N
ENABLED_FLAG	NOT NULL	VARCHAR2(1)	Y
VENDOR_ID		NUMBER	Foreign key: PO_VENDORS
VENDOR_SITE_ID		NUMBER	Foreign key: PO_VENDOR_SITES

Table 7 – 6 PO_HEADERS_ALL (Page 1 of 2)

Column Name	Null	Type	Comments
TERMS_ID		NUMBER	
FREIGHT_TERMS_LOOKUP_CODE		VARCHAR2(25)	Foreign key: PO_LOOKUP_CODES
CURRENCY_CODE		VARCHAR2(15)	
APPROVED_FLAG		VARCHAR2(1)	Y
ORG_ID		NUMBER	Multiple Organization Support

Table 7 – 6 PO_HEADERS_ALL (Page 2 of 2)

AGENT_ID

Enter the ID for the agent who created the purchase order.

This value is used by the following reports in your Oracle Payables application: Vendor Merge, Matching Agent Notice and Receiving Hold Requestor Notice.

TYPE_LOOKUP_CODE

Enter BLANKET or STANDARD (Lookup Type: PO TYPE) to identify the type of purchase order.

TERMS_ID

Enter the ID for the payment terms of the purchase order. Your Oracle Payables application uses this value during matching to warn you if the payment terms on the purchase order do not match the payment terms on the invoice being matched to the purchase order.

If you choose to leave this column empty, your Oracle Payables application will not warn you if the purchase order and invoice payment terms differ.

FREIGHT_TERMS_LOOKUP_CODE

Enter a QuickCode (QuickCode Type: FREIGHT TERMS) to identify the freight terms for the purchase order.

CURRENCY_CODE

Enter the currency code for the purchase order. You can obtain a list of valid codes from FND_CURRENCIES.CURRENCY_CODE.

The currency code for the invoice you want to match to this purchase order must be the same as the code you enter here.

ORG_ID

If you use the Oracle Applications multiple organizations feature, enter the operating unit's org_id. If you do not use the Oracle Applications multiple organizations feature, leave this column null. To insert rows into open interface tables, use the multiple organization PO_HEADERS_ALL table and explicitly populate the org_id column. You can determine an operating unit's org_id by querying the HR_OPERATING_UNITS view:

```
SELECT name, organization_id from  
HR_OPERATING_UNITS;
```

When you import rows *from* open interface tables, only rows from your current operating unit are processed.

PO_RELEASES_ALL (Blanket releases)

Column Name	Null	Type	Comments
PO_RELEASE_ID	NOT NULL	NUMBER	Primary key
LAST_UPDATE_DATE	NOT NULL	DATE	
LAST_UPDATED_BY	NOT NULL	NUMBER	
PO_HEADER_ID	NOT NULL	NUMBER	Foreign key: PO_HEADERS
RELEASE_NUM	NOT NULL	NUMBER	Release number
AGENT_ID	NOT NULL	NUMBER	Foreign key: HR_EMPLOYEES
RELEASE_DATE	NOT NULL	DATE	
ORG_ID		NUMBER	Multiple Organiza- tion Support

Table 7 – 7 PO_RELEASES_ALL (Page 1 of 1)

PO_LINES_ALL

Column Name	Null	Type	Comments
PO_LINE_ID	NOT NULL	NUMBER	Primary key
LAST_UPDATE_DATE	NOT NULL	DATE	
LAST_UPDATED_BY	NOT NULL	NUMBER	
PO_HEADER_ID	NOT NULL	NUMBER	Foreign key: PO_HEADERS
LINE_TYPE_ID	NOT NULL	NUMBER	Foreign key: PO_LINE_TYPES
LINE_NUM		NUMBER	Used for Query
ITEM_ID		NUMBER	Null
ITEM_DESCRIPTION		VARCHAR2(240)	Display only
UNIT_PRICE		NUMBER	
TYPE_1099		VARCHAR2(10)	
ORG_ID		NUMBER	Multiple Organiza- tion Support

Table 7 – 8 PO_LINES_ALL (Page 1 of 1)

LINE_TYPE_ID

Enter the ID for the line type of the purchase order line. You can obtain a list of valid IDs from PO_LINE_TYPES.LINE_TYPE_ID.

ITEM_ID

Do not enter a value in this column. Your Oracle Payables application does not allow you to record purchase order lines with Items unless you install Oracle Purchasing or Oracle Government Purchasing.

ITEM_DESCRIPTION

Enter a description for your purchase order line. You can use this column to record information about the item on the purchase order line. Your Oracle Payables application displays this description in the Purchase Order Shipments zone during matching.

TYPE_1099

Enter the income tax type for the purchase order line, if the vendor for the purchase order is a 1099 vendor. Your Oracle Payables application

assigns this type as the default income tax type for each invoice distribution line created by matching to this purchase order line. If you leave the column empty, your Oracle Payables application uses the income tax type for the vendor as the default.

You can obtain a list of valid types from
AP_INCOME_TAX_TYPES.INCOME_TAX_TYPE.

ORG_ID

If you use the Oracle Applications multiple organizations feature, enter the operating unit's org_id. If you do not use the Oracle Applications multiple organizations feature, leave this column null. To insert rows into open interface tables, use the multiple organization PO_LINES_ALL table and explicitly populate the org_id column. You can determine an operating unit's org_id by querying the HR_OPERATING_UNITS view:

```
SELECT name, organization_id from
HR_OPERATING_UNITS;
```

When you import rows *from* open interface tables, only rows from your current operating unit are processed.

PO_LINE_LOCATIONS_ALL (PO Shipments)

Column Name	Null	Type	Comments
LINE_LOCATION_ID	NOT NULL	NUMBER	Primary key
LAST_UPDATE_DATE	NOT NULL	DATE	
LAST_UPDATED_BY	NOT NULL	NUMBER	
PO_HEADER_ID	NOT NULL	NUMBER	Foreign key: PO_HEADERS
PO_LINE_ID	NOT NULL	NUMBER	Foreign key: PO_LINES
QUANTITY		NUMBER	Quantity ordered: 2-way match
QUANTITY_RECEIVED		NUMBER	3-way match
QUANTITY_ACCEPTED		NUMBER	4-way match
QUANTITY_BILLED		NUMBER	UPDATED by Payables
QUANTITY_CANCELLED		NUMBER	0 or Null

Table 7 – 9 PO_LINE_LOCATIONS_ALL (Page 1 of 2)

Column Name	Null	Type	Comments
UNIT_MEAS_LOOKUP_CODE		VARCHAR2(25)	Display only
PO_RELEASE_ID		NUMBER	Foreign key: PO_RELEASES
SHIP_TO_LOCATION_ID		NUMBER	Foreign key: HR_LOCATIONS
TAXABLE_FLAG		VARCHAR2(1)	Y or N (tax matching)
TAX_NAME		VARCHAR2(15)	Tax name match (TAX DIFF HOLD)
APPROVED_FLAG		VARCHAR2(1)	Y
SHIPMENT_NUM		NUMBER	Shipment number
SHIPMENT_TYPE		VARCHAR2(25)	Foreign Key: PO_LOOKUP_CODES
INSPECTION_REQUIRED_FLAG		VARCHAR2(1)	Y or N (4-way match)
RECEIPT_REQUIRED_FLAG		VARCHAR2(1)	Y or N (3-way match)
CLOSED_CODE		VARCHAR2(1)	Null
ORG_ID		NUMBER	Multiple Organization Support

Table 7 – 9 PO_LINE_LOCATIONS_ALL (Page 2 of 2)

QUANTITY

Enter the quantity of goods ordered for the purchase order shipment. Your Oracle Payables application uses this amount to match against if you are using 2-way matching. Your Oracle Payables application verifies that this quantity matches the invoice quantity within defined tolerance levels and places the invoice on hold if it doesn't match. In addition, if the quantity of the invoice is greater than the shipment quantity, your Oracle Payables warns you during invoice entry that the match will result in an overbill.

QUANTITY_RECEIVED/QUANTITY_ACCEPTED

Enter the quantity of goods received/accepted if you are using 3-way/4-way matching. Your Oracle Payables application verifies that the quantity matches the invoice quantity within defined tolerance levels and places the invoice on hold if it doesn't match.

QUANTITY_BILLED

Do not enter a value in this column, unless you have already matched an invoice to this purchase order shipment. When a match successfully completes (invoice is approved), your Oracle Payables application updates this column with the quantity you specified during matching.

QUANTITY_CANCELLED

Only enter a value in this column if you have cancelled a portion of the purchase order shipment in your non-Oracle purchasing system. The amount you enter reduces the amount that Oracle Payables considers to be the outstanding quantity ordered. Oracle Payables displays a warning if you try to match to a shipment which has been cancelled.

When you cancel a shipment, your Oracle Purchasing application sets the PO_LINE_LOCATIONS_ALL.QUANTITY_CANCELLED to:

- $QUANTITY - QUANTITY_RECEIVED$ if receipt is required
- $QUANTITY - QUANTITY_BILLED$ if receipt is not required.



Attention: AutoApproval does all quantity checks assuming the QUANTITY is the actual QUANTITY minus the QUANTITY_CANCELLED.

UNIT_MEAS_LOOKUP_CODE

Oracle Payables displays this value in the matching zones, but does not validate the column. You can enter any value into this column; however, you should use the same value that you use in your non-Oracle purchasing system.

TAXABLE_FLAG

Enter Y or N to indicate the purchase order shipment is subject to tax. If you enter Y, enter a value in the TAX_NAME column. During AutoApproval, your Oracle Payables application verifies that the tax name for the purchase order shipment matches the tax name on the invoice and places a Tax Difference hold on the invoice if the tax names don't match.

TAX_NAME

Enter the tax name used to verify that the tax names on the invoice and purchase order shipment match. You do not need to enter a value if you enter N in the TAXABLE_FLAG column.

You can obtain a list of valid tax names from
AP_TAX_CODES_ALL.NAME.

TYPE_LOOKUP_CODE

Enter BLANKET, STANDARD, or SCHEDULED (Lookup Type: SHIPMENT TYPE) to identify the type of purchase order shipment.

CLOSED_CODE

Do not enter a value in this column if you wish to match an invoice to this purchase order shipment. If you enter the values CLOSED, FINALLY CLOSED, or CLOSED FOR INVOICE in the column, your Oracle Payables application warns you that you are matching to a closed purchase order.

ORG_ID

If you use the Oracle Applications multiple organizations feature, enter the operating unit's org_id. If you do not use the Oracle Applications multiple organizations feature, leave this column null. To insert rows into open interface tables, use the multiple organization PO_LINE_LOCATIONS_ALL table and explicitly populate the org_id column. You can determine an operating unit's org_id by querying the HR_OPERATING_UNITS view:

SELECT name, organization_id from
HR_OPERATING_UNITS;

When you import rows *from* open interface tables, only rows from your current operating unit are processed.

PO_DISTRIBUTIONS_ALL (Account distribution)

Your Oracle Payables application uses a view (PO_DISTRIBUTIONS_AP_V) to this table to perform purchase order distribution line matching.

Column Name	Null	Type	Comments
PO_DISTRIBUTION_ID	NOT NULL	NUMBER	Primary key
LAST_UPDATE_DATE	NOT NULL	DATE	
LAST_UPDATED_BY	NOT NULL	NUMBER	

Table 7 – 10 PO_DISTRIBUTIONS_ALL (Page 1 of 3)

Column Name	Null	Type	Comments
PO_HEADER_ID	NOT NULL	NUMBER	Foreign key: PO_HEADERS
PO_LINE_ID	NOT NULL	NUMBER	Foreign key: PO_LINES
LINE_LOCATION_ID	NOT NULL	NUMBER	Foreign key: PO_LINE_ LOCATIONS
SET_OF_BOOKS_ID	NOT NULL	NUMBER	
CODE_ COMBINATION_ID		NUMBER	Foreign key: GL_CODE_ COMBINATIONS
QUANTITY_ORDERED		NUMBER	
QUANTITY_DELIVERED		NUMBER	Display only
QUANTITY_BILLED		NUMBER	UPDATED by Payables
QUANTITY_CANCELLED		NUMBER	0 or Null
DELIVER_TO_ PERSON_ID		NUMBER	Foreign key: HR_EMPLOYEES
RATE		NUMBER	Foreign currency
AMOUNT_BILLED		NUMBER	UPDATED by Payables
ENCUMBERED_AMOUNT		NUMBER	If encumbrance accounting enabled
BUDGET_ACCOUNT_ID		NUMBER	
ACCRUAL_ACCOUNT_ID		NUMBER	
VARIANCE_ACCOUNT_ID		NUMBER	
DISTRIBUTION_NUM	NOT NULL	NUMBER	Distribution line number
PROJECT_ID		NUMBER	Project Accounting
TASK_ID		NUMBER	Project Accounting
EXPENDITURE_TYPE		VARCHAR2(30)	Project Accounting
PROJECT_ ACCOUNTING_CONTEXT		VARCHAR2(30)	Project Accounting
EXPENDITURE_ ORGANIZATION_ID		NUMBER	Project Accounting

Table 7 – 10 PO_DISTRIBUTIONS_ALL (Page 2 of 3)

Column Name	Null	Type	Comments
EXPENDITURE_ITEM_DATE		DATE	Project Accounting
ORG_ID		NUMBER	Multiple Organization Support

Table 7 – 10 PO_DISTRIBUTIONS_ALL (Page 3 of 3)

SET_OF_BOOKS_ID

Enter the set of books ID for your purchase order distribution line. The ID you enter must be for the set of books you define in the Define Financials Options form.

CODE_COMBINATION_ID

Enter the Accounting Flexfield ID for the expense account you want to charge for the goods on the purchase order distribution line.

QUANTITY_ORDERED

Enter the amount of goods charged to the Accounting Flexfield for this purchase order distribution line.



Attention: NOTE: Oracle Payables does not validate the following, but assumes it to be true:

Total of PO_DISTRIBUTIONS_ALL.QUANTITY_ORDERED
for one PO_LINE_LOCATION_ID =
PO_LINES_LOCATIONS.QUANTITY (for the same ID).

Oracle Payables sometimes prorates the
PO_DISTRIBUTION.QUANTITY_ORDERED using the
PO_LINES_LOCATIONS.QUANTITY as the total.

BUDGET_ACCOUNT_ID/ACCRUAL_ACCOUNT_ID/ VARIANCE_ACCOUNT_ID

Enter the same Accounting Flexfield ID you entered for the CODE_COMBINATION_ID. Your Oracle Payables application allows you to record budget, accrual, and variance (price and exchange rate) amounts for your purchase order distribution lines, but requires you to charge these amounts to the same expense account for the distribution line.

QUANTITY_BILLED

Do not enter a value in this column, unless you have already matched an invoice distribution line to this purchase order distribution line. When a match successfully completes (invoice is approved), your Oracle Payables application updates this column with the quantity you specified during matching.

QUANTITY_CANCELLED

Only enter a value in this column if you have cancelled a portion of the purchase order distribution in your non-Oracle purchasing system. The amount you enter reduces the amount that Oracle Payables considers to be the outstanding quantity ordered. Oracle Payables displays a warning if you try to match to a shipment which has been cancelled.



Attention: AutoApproval does all quantity checks assuming the quantity for the distribution is `QUANTITY_ORDERED` minus `QUANTITY_CANCELLED`.

AMOUNT_BILLED

Do not enter a value in this column, unless you have already matched an invoice to this purchase order shipment. When a match successfully completes (invoice is approved), your Oracle Payables application updates this column with the amount of the quantity you specified during matching multiplied by the unit price.

Project Accounting Columns

Enter project information from Oracle Project Accounting if you want to associate the invoice distribution line (created through matching) with a project in Oracle Project Accounting. Your Oracle Payables application transfers the information into the `AP_INVOICE_DISTRIBUTIONS_ALL` table and uses it to create the default Accounting Flexfield for the invoice distribution line.

ORG_ID

If you use the Oracle Applications multiple organizations feature, enter the operating unit's `org_id`. If you do not use the Oracle Applications multiple organizations feature, leave this column null. To insert rows into open interface tables, use the multiple organization `PO_DISTRIBUTIONS_ALL` table and explicitly populate the `org_id`

column. You can determine an operating unit's org_id by querying the HR_OPERATING_UNITS view:

```
SELECT name, organization_id from  
HR_OPERATING_UNITS;
```

When you import rows *from* open interface tables, only rows from your current operating unit are processed.

Matching to Purchase Orders

Perform the following steps to match invoices to purchase order information from your non-Oracle purchasing system:

- Create flat file with purchasing information
- Load information into purchasing tables
- Enter invoice(s)
- Match invoice(s) to purchase order shipment or purchase order distribution
- Submit AutoApproval for invoice(s)

Create Flat File with Purchasing Information

To load invoice information into your Oracle Payables application via SQL*Loader, first create a program that produces a flat file containing the information from your non-Oracle purchasing system for the purchase orders you want to match to invoices.

Load Information into Purchasing Tables

Use SQL*Loader to load the required information into the purchasing tables. You will need to create a SQL*Loader control file to format the information you want to load. The file you write will vary greatly depending on the nature and format of the flat file you use. Your control file must populate the purchasing tables as indicated in the previous table descriptions.



Additional Information: SQL*Loader
(*ORACLE7 Server Utilities Guide*)

Enter Invoices

You match invoices to purchase order shipments during invoice entry. This online function links an invoice in the database to one or more purchase order shipments you choose. You cannot pay or post an invoice until AutoApproval approves the invoice.

You can match any type of invoice to a purchase order, including credit and debit memos. Your Oracle Payables application provides you with two invoice types, QuickMatch and PO Default, that you can use to quickly and easily match an invoice to a purchase order.



Overview of Invoices

(*Oracle Payables Reference Manual* or *Oracle Government Payables Reference Manual*)



Enter Invoices

Adjust Distributions

(*Oracle Payables Reference Manual* or *Oracle Government Payables Reference Manual*)

Match to Purchase Order Shipments and Distribution Lines

When you choose the matching option during invoice entry, your Oracle Payables application displays purchase order shipments for the vendor. You indicate whether you want to match to the purchase order shipment or to specific invoice distribution lines. You then choose the shipment or distribution line you want to match to, and the quantity and price you are matching. Then your Oracle Payables application performs the following for each matched shipment:

- Update QUANTITY_BILLED and AMOUNT_BILLED in PO_DISTRIBUTIONS_ALL
- Update QUANTITY_BILLED in PO_LINE_LOCATIONS_ALL
- Create one or more AP_INVOICE_DISTRIBUTIONS_ALL which record the QUANTITY_INVOICED, UNIT_PRICE, and PO_DISTRIBUTION_ID, in addition to other Payables information.



Purchasing Integration

(*Oracle Payables Reference Manual* or *Oracle Government Payables Reference Manual*)

Match to Credit and Debit Memos

Your Oracle Payables application lets you enter a credit or debit memo (with a negative amount) and match to a purchase order. You would enter a negative quantity in the Quantity Invoiced field in the Purchase Order Shipment Match zone, thereby matching this credit invoice to one or no purchase order shipment lines. Your Oracle Payables application then decreases the quantity billed against the purchase order shipment line(s).

When you match a credit invoice to a purchase order shipment line, your Oracle Payables application:

- Reopens closed shipment lines (sets PO_LINE_LOCATIONS_ALL.CLOSED_CODE to NULL)
- Updates PO_LINE_LOCATIONS_ALL.QUANTITY_BILLED
- Updates PO_DISTRIBUTIONS_ALL.QUANTITY_BILLED



Attention: Your Oracle Payables application does not update any receiving information. You must install an Oracle Purchasing application if you want to enter or update receiving information for a purchase order

Close a Purchase Order Shipment

Invoice entry closes a purchase order shipment (sets CLOSED_CODE in PO_LINE_LOCATIONS_ALL to 'CLOSED') when:

- QUANTITY_BILLED equals or exceeds QUANTITY_ORDERED (two-way matching), or
- QUANTITY_ORDERED is less than or equal to QUANTITY_RECEIVED and QUANTITY_RECEIVED is less than or equal to QUANTITY_BILLED

Final Close

Your Oracle Payables application does not support finally closing a purchase order if you do not install an Oracle Purchasing application with your Oracle Payables application. Final close allows you to match an invoice to a purchase order and permanently close the purchase order when you approve the invoice.



Purchasing Integration
(*Oracle Payables Reference Manual* or *Oracle Government Payables Reference Manual*)

Online Review of Purchasing Information

Without an Oracle Purchasing application, your Oracle Payables application does not allow you to review purchasing information, such as purchase order header and line information, online in the invoice forms (Enter Invoices, Adjust Distributions, and View Invoices).



Enter Invoices

Adjust Distributions

View Invoices

(Oracle Payables Reference Manual or Oracle Government Payables Reference Manual)

Using AutoApproval

AutoApproval is the Oracle Payables feature that performs two-, three-, or four-way matching. An invoice must pass AutoApproval before you can pay or post the invoice. AutoApproval reviews each invoice and places one or more matching holds on the invoice if the invoice does not meet your matching criteria. It also releases any existing matching holds if you adjust your invoice or purchase order to meet your matching criteria and current information on order, receipt and acceptance prices and quantities.

You must submit AutoApproval for all invoices, not just matched invoices, since it also checks for distribution variances, tax variances, and exchange rate information. You can submit AutoApproval online for an invoice or in batch for a group of invoices.



Invoice Approval

(Oracle Payables Reference Manual or Oracle Government Payables Reference Manual)



Submit AutoApproval Process

(Oracle Payables Reference Manual or Oracle Government Payables Reference Manual)

2-way, 3-way, and 4-way Matching

When you match to a purchase order, your Oracle Payables application automatically checks that the total of

PO_DISTRIBUTIONS_ALL.QUANTITY_ORDERED =

AP_INVOICE_DISTRIBUTIONS_ALL.QUANTITY_INVOICED (2-way matching).

Your Oracle Payables application only checks QUANTITY_RECEIVED (3-way matching) if the RECEIPT_REQUIRED_FLAG is set to Y and only checks QUANTITY_ACCEPTED (4-way matching) if the INSPECTION_REQUIRED_FLAG is set to Y.

Tax Matching

Your Oracle Payables application only checks for tax name matching if the system option MATCH ON TAX is set to Y and the invoice has distribution lines with tax names.

Matching Tolerance

You can define percentage and amount tolerances for Matching quantities and price. Your Oracle Payables application places a matching hold on an invoice only if the invoice quantity or price is greater than the purchasing quantity or price by more than your tolerance.

Matching Holds

When you submit AutoApproval, your Oracle Payables application places a matching hold on a matched invoice (by inserting one or more rows in AP_HOLDS_ALL, one row for each type of hold for each invoice distribution) if:

- QUANTITY_BILLED > QUANTITY in PO_LINE_LOCATIONS_ALL (QTY ORD Hold)
- UNIT_PRICE in AP_INVOICE_DISTRIBUTIONS_ALL > PRICE_OVERRIDE in PO_LINE_LOCATIONS_ALL (PRICE Hold)
- QUANTITY_BILLED > QUANTITY_RECEIVED in PO_LINE_LOCATIONS_ALL (QTY REC Hold)
- QUANTITY_BILLED > QUANTITY_ACCEPTED in PO_LINE_LOCATIONS_ALL (QUALITY Hold)
- TAXABLE_FLAG = NO in PO_LINE_LOCATIONS_ALL, but there IS tax recorded on the invoice (TAX DIFFERENCE Hold)
- TAX_NAME in PO_LINE_LOCATIONS_ALL is not equal to VAT_CODE in AP_INVOICE_DISTRIBUTIONS_ALL (TAX DIFFERENCE Hold)

Release Matching Holds

When you submit AutoApproval, your Oracle Payables application also releases a matching hold if any of the above conditions are corrected by:

- Updating the order quantity or price of the shipment
- Updating the received quantity for the shipment
- Updating the accepted quantity for the shipment
- Updating the taxable flag or tax name on the shipment

Your Oracle Payables application stores detailed information for both matching holds and releases in AP_HOLDS_ALL.

Invoice Approval (Manual Release)

You can also manually release most matching holds in your Oracle Payables application if you need to pay an invoice which does not pass your matching criteria.



Invoice Approval

(*Oracle Payables Reference Manual* or *Oracle Government Payables Reference Manual*)

Using Encumbrance Accounting

Your Oracle Payables application supports using encumbrance accounting with a non-Oracle purchasing system. To use encumbrance accounting, however, you must initially record the encumbered amount for the purchase order to which you want to match an invoice.

Then, when AutoApproval approves the invoice, if there is a variance between the invoice and its matched purchase order within the tolerances you define, your Oracle Payables application automatically creates an encumbrance journal entry for the amount of the variance. Your Oracle Payables application always creates encumbrance journal entries in detail.

With an Oracle Purchasing application installed, your Oracle Payables application allows you to record these variance encumbrance journal entries to a separate variance account. With a non-Oracle purchasing system, your Oracle Payables application requires you to record the variance amount to the same Accounting Flexfield as the expense Accounting Flexfield for the purchase order distribution line.

When you post the invoice to your general ledger, your Oracle Payables application relieves both the original encumbrance journal entries that you created when you encumbered the purchase order and the encumbrance journal entries it automatically created for the variance. Your Oracle Payables application then creates actual journal entries for your invoice transaction. Your variance encumbrance journal entries and your actual journal entries update your account balances only when you post the journal entries in your general ledger.

You enable encumbrance accounting using the Define Financials Options form.



Define Financials Options

(Oracle Payables Reference Manual or Oracle Government Payables Reference Manual)

Budgetary Control

Your Oracle Payables application does not allow you to use the budgetary control feature if you do not install an Oracle Purchasing application with your application.

Purging Purchasing Information

Your Oracle Payables application does not allow you to purge purchasing information if you do not have an Oracle Purchasing application installed. When you match an invoice to a purchase order from a non-Oracle purchasing system, you will not be able to purge the invoice because your Oracle Payables application requires that all objects, including matched purchase orders, associated with an invoice must be purgeable before you can purge the invoice.



Purge

(Oracle Payables Reference Manual or Oracle Government Payables Reference Manual)

Index

A

ACE interface

- FA_ACE_BOOKS table, 2 – 54
- loading ACE table, 2 – 56
- manual loading, 2 – 59
- overview, 2 – 53
- update process, 2 – 54

AutoClear

- AP_RECON_DETAILS table, 7 – 8
- AP_RECON_SUMMARIES table, 7 – 9
- definitions, 7 – 5
- major features, 7 – 3
- overview, 7 – 2
- reconciling payment information, 7 – 17
- setting up, 7 – 10
- tables, 7 – 7
- using, 7 – 6

AutoInvoice

- account assignments, 10 – 31
- accounting rules, 6 – 18
- credit memos, 10 – 43
- customer bank accounts, 10 – 22
- deriving dates, 10 – 45
- finance charges, 10 – 31
- freight lines, 10 – 23
- grouping rules, 10 – 38
- integrating Order Entry with Receivables, 6 – 15
- invoices, 10 – 42
- invoicing rules, 6 – 18
- line ordering rules, 10 – 41
- overview, 10 – 2

payment methods, 10 – 22

preparations, 10 – 9

RA_INTERFACE_DIST table, 10 – 102

RA_INTERFACE_LINES table, 6 – 24,
10 – 51

RA_INTERFACE_SALESCREDIT table,
6 – 45

RA_INTERFACE_SALESCREDITS table,
10 – 98

setup data, 10 – 15

tax, 10 – 24

transaction flexfields, 10 – 33

transferring data, 10 – 18

using, 10 – 20

validation, 10 – 19

AutoLockbox

AR_PAYMENTS_INTERFACE table,
10 – 122

importing data, 10 – 116, 10 – 122

overview, 10 – 109

preparations, 10 – 116

questions, 10 – 140

reviewing receipts, 10 – 121

updating balance, 10 – 121

validation, 10 – 117

B

Budget interface

entering information, 2 – 41

FA_BUDGET_INTERFACE table, 2 – 42

overview, 2 – 38

prerequisites, 2 – 39

SQL Loader script, 2 – 43

Budget upload

- GL_BUDGET_INTERFACE table, 4 – 36
- overview, 4 – 33
- prerequisites, 4 – 35
- running, 4 – 45
- status report, 4 – 45
- transferring data, 4 – 44
- uploading budget information, 4 – 42

C

Creating assets from invoices. *See* Mass Additions

Customer interface

- generating original system references, 10 – 197
- importing data, 10 – 150
- overview, 10 – 143
- preparations, 10 – 149
- RA_CONTACT_PHONES_INTERFACE table, 10 – 181
- RA_CUST_PAY_METHOD_INTERFACE table, 10 – 192
- RA_CUSTOMER_BANKS_INTERFACE table, 10 – 185
- RA_CUSTOMER_PROFILES_INTERFACE table, 10 – 167
- RA_CUSTOMERS_INTERFACE table, 10 – 158
- relationships (valid), 10 – 195
- using, 10 – 156, 10 – 157

D

Demand interface

- ATP check, 5 – 12
- ATS query, 5 – 12
- ATS Quick Pick, 5 – 12
- demand add, 5 – 7
- demand and reservation add, 5 – 10
- function descriptions, 5 – 6
- MTL_DEMAND_INTERFACE table, 5 – 14
- overview, 5 – 2
- pick release, 5 – 11

- reservation add, 5 – 9
- reservation modify, 5 – 10
- resolving failed rows, 5 – 21
- setting up, 5 – 4
- update forecast attributes, 5 – 10
- validation, 5 – 20

I

Importing orders. *See* OrderImport

Importing transactions. *See* Transaction Import

Interface tables

- AP_EXPENSE_REPORT_HEADERS, 7 – 25
- AP_EXPENSE_REPORT_LINES, 7 – 37
- AP_INVOICE_DISTRIBUTIONS, 7 – 51
- AP_INVOICES, 7 – 51
- AP_RECON_DETAILS, 7 – 8
- AP_RECON_SUMMARIES, 7 – 9
- AR_PAYMENTS_INTERFACE, 10 – 122
- FA_ACE_BOOKS, 2 – 54
- FA_BUDGET_INTERFACE, 2 – 42
- FA_MASS_ADDITONS, 2 – 20
- FA_PRODUCTION_INTERFACE, 2 – 50
- GL_BUDGET_INTERFACE, 4 – 36
- GL_INTERFACE, 4 – 9
- MTL_DEMAND_INTERFACE, 5 – 14
- MTL_REPLENISH_HEADERS_INT, 5 – 49
- MTL_SERIAL_NUMBERS_INTERFACE, 5 – 42
- MTL_TRANSACTION_LOTS_INTERFACE, 5 – 41
- MTL_TRANSACTIONS_INTERFACE, 5 – 28
- PA_TRANSACTION_INTERFACE, 8 – 7
- PO_DISTRIBUTIONS, 7 – 59
- PO_HEADERS, 7 – 52
- PO_LINE_LOCATIONS, 7 – 56
- PO_LINES, 7 – 55
- PO_RELEASES, 7 – 54
- PO_REQUISITIONS_INTERFACE, 9 – 6
- RA_CONTACT_PHONES_INTERFACE, 10 – 181
- RA_CUST_PAY_METHOD_INTERFACE, 10 – 192

- RA_CUSTOMER_BANKS_INTERFACE, 10 – 185
- RA_CUSTOMER_PROFILES_INTERFACE, 10 – 167
- RA_CUSTOMERS_INTERFACE, 10 – 158
- RA_INTERFACE_DIST, 10 – 102
- RA_INTERFACE_LINES, 6 – 24, 10 – 51
- RA_INTERFACE_SALESCREDITS, 10 – 98
- SO_HEADERS_INTERFACE, 6 – 72
- SO_LINE_DETAILS_INTERFACE, 6 – 118
- SO_LINES_INTERFACE, 6 – 99
- SO_PRICE_ADJUSTMENTS_INTERFACE, 6 – 124
- SO_SALES_CREDITS_INTERFACE, 6 – 130

Internal Sales Orders, not interfaced to Receivables, 6 – 20

Invoice import

- AP_EXPENSE_REPORT_HEADERS table, 7 – 25

- AP_EXPENSE_REPORT_LINES table, 7 – 37

- definitions, 7 – 23

- major features, 7 – 21

- overview, 7 – 20

- setting up, 7 – 46

Invoice information. *See* AutoInvoice

Invoices, importing from other systems, 7 – 20

J

Journal import

- budget data, 4 – 23

- encumbrance data, 4 – 23

- GL_INTERFACE table, 4 – 9

- importing data, 4 – 7

- importing from feeder system, 1 – 4

- importing specialized data, 4 – 22

- intercompany data, 4 – 23

- major features, 4 – 3

- multi-currency data, 4 – 22

- overview, 4 – 2

- preparing Oracle General Ledger, 4 – 26

- prerequisites, 4 – 7

- specifying Accounting Flexfields, 4 – 11

- statistical data, 4 – 23

- using, 4 – 28

- validation, 4 – 24

M

Mass additions

- defining, 2 – 10

- FA_MASS_ADDITIONS table, 2 – 20

- finishing import, 2 – 35

- importing information, 2 – 19

- importing steps, 2 – 5

- loading data, 2 – 16

- overview, 2 – 2

- planning import, 2 – 7

Matching. *See* Purchase orders

Multiple Organization Support

- AutoInvoice, 10 – 69, 10 – 100, 10 – 105, 10 – 108, 10 – 132

- importing invoices, 7 – 34, 7 – 42

O

Open demand interface. *See* Demand interface

Open interfaces

- importing information into Oracle Financials, 1 – 2

- overview, 1 – 1

Open replenishment interface. *See* replenishment interface

Open requisitions

- derived data, 9 – 17

- optional data, 9 – 18

- overview, 9 – 3

- PO_REQUISITONS_INTERFACE table, 9 – 6

- required data, 9 – 16

- resolving failed rows, 9 – 20

- setup, 9 – 5

- validation, 9 – 19

Open transaction interface. *See* Transaction interface

- Oracle Assets
 - ACE interface, 2 – 53
 - budget interface, 2 – 38
 - converting asset information, 2 – 2
 - mass additions interface, 2 – 2
 - production interface, 2 – 46
- Oracle General Ledger
 - budget upload, 4 – 33
 - journal import, 4 – 2
- Oracle Inventory
 - Open demand interface, 5 – 22
 - open demand interface, 5 – 2
 - open item interface, 5 – 58
 - Open replenishment interface, 5 – 46
- Oracle Order Entry
 - interfacing with Receivables, 6 – 15
 - OrderImport, 6 – 48
- Oracle Order Entry / Shipping, Ship Confirm
 - Open Interface, 6 – 2
- Oracle Payables
 - integrating with Oracle Purchasing, 9 – 23
 - purge, 9 – 23
- Oracle Payables Applications
 - AutoClear, 7 – 2
 - integrating purchase orders, 7 – 50
 - Invoice import, 7 – 20
- Oracle Purchasing
 - integrating with Oracle Payables, 9 – 23
 - open requisitions interface, 9 – 2
 - purge, 9 – 23
- Oracle Receivables
 - interfacing with Order Entry, 6 – 15
 - sales interface, 10 – 198
- Oracle Receivables Applications
 - AutoInvoice, 10 – 2
 - AutoLockbox, 10 – 109
 - customer interface, 10 – 143
 - Sales Tax Rate Interface, 10 – 198
 - Tax Vendor Extension, 10 – 219
- OrderImport
 - changes to imported orders, 6 – 61
 - complete orders, 6 – 49, 6 – 70
 - configurations, 6 – 59
 - credit checking, 6 – 53
 - defaulting, 6 – 53, 6 – 68, 6 – 70
 - entered state, 6 – 49
 - holds, 6 – 53
 - importing data, 6 – 57
 - internal sales orders, 6 – 53
 - items, 6 – 58
 - not null columns, 6 – 59
 - notes, 6 – 53
 - operation code, 6 – 61
 - order scheduling, 6 – 50
 - overview, 6 – 48
 - prerequisites, 6 – 55
 - price adjustments, 6 – 50, 6 – 51, 6 – 71
 - pricing, 6 – 50, 6 – 51
 - reporting, 6 – 54
 - sales credits, 6 – 52, 6 – 70
 - schedule details, 6 – 50, 6 – 60, 6 – 71
 - security rules, 6 – 53, 6 – 70
 - ship sets, 6 – 61
 - shipment schedules, 6 – 60
 - shipping complete orders, 6 – 61
 - standard values, 6 – 53, 6 – 68, 6 – 70
 - using IDs, 6 – 66
 - using multi-currency, 6 – 61
 - validation, 6 – 68 to 6 – 71
- Other Information Sources, vi

P

- Production interface
 - entering information, 2 – 49
 - FA_PRODUCTION_INTERFACE table, 2 – 50
 - overview, 2 – 46
 - SQL Loader script, 2 – 51
- Purchase orders
 - AP_INVOICE_DISTRIBUTIONS table, 7 – 51
 - AP_INVOICES table, 7 – 51
 - encumbrance accounting, 7 – 68
 - integrating with Oracle Payables, 7 – 50
 - matching, 7 – 63
 - overview, 7 – 50

PO_DISTRIBUTIONS table, 7 – 59
PO_HEADERS table, 7 – 52
PO_LINE_LOCATIONS, 7 – 56
PO_LINES table, 7 – 55
PO_RELEASES table, 7 – 54
purging information, 7 – 69
understanding matching tables, 7 – 50
using AutoApproval, 7 – 66

Purge

overview, 9 – 23
tables, 9 – 23

R

Replenishment interface

fixing failed transactions, 5 – 57
MTL_REPLENISH_HEADERS_INT table,
5 – 49
overview, 5 – 46
setting up, 5 – 48
validation, 5 – 55
viewing failed transactions, 5 – 56

ReqImport, rescheduling requisitions, 9 – 21

S

Sales interface

action codes, 10 – 203
city names, 10 – 214
date adjustments, 10 – 209
overview, 10 – 198
preparations, 10 – 200
program updates, 10 – 214
report termination, 10 – 215
SQL Loader script, 10 – 215, 10 – 216
transferring data, 10 – 202

Ship Confirm Interface Tables,

SHP_PICKING_HEADERS_INTERFACE,
6 – 5, 6 – 8

Ship Confirm Interface tables,

SHP_PICKING_HEADERS_INTERFACE,
6 – 11

Ship Confirm Open Interface

See also replenishment interface
fixing failed transactions, 6 – 13
functional overview, 6 – 2
Inserting into Shipping Tables, 6 – 5
Shipping Transaction Manager, 6 – 13
SHP_PICKING_HEADERS_INTERFACE
table, 6 – 5
SHP_PICKING_LINE_DETAILS
INTERFACE_V table, 6 – 8, 6 – 11
transactions, 6 – 3
validation, 6 – 12
viewing failed transactions, 6 – 13

Shipping Transaction Manager, 6 – 13

Spreadsheet interface

budget upload, 4 – 33
entering data, 4 – 38
interface table, 4 – 36
major features, 4 – 33
reporting data, 4 – 47
reviewing status report, 4 – 45
software requirements, 4 – 43
spreadsheet contents, 4 – 37
transferring data, 4 – 44
using, 4 – 33
validating data, 4 – 45

T

Tax. *See* sales interface

Tax Vendor Hook, overview, 10 – 219

Transaction Import, 8 – 2

correcting rejected transactions, 8 – 18
interface control table, 8 – 6
interface table, 8 – 4
not null columns, 8 – 5
null columns, 8 – 5
overview, 8 – 2
populating interface table, 8 – 14
purging transactions, 8 – 14
resolving exceptions, 8 – 16
running, 8 – 15
system assigned columns, 8 – 6
table descriptions, 8 – 6

transaction sources, 8 – 14	overview, 5 – 22
validation, 8 – 2 to 8 – 4	resolving failed rows, 5 – 44
viewing successful transactions, 8 – 16	setting up, 5 – 26
Transaction interface	validation, 5 – 44
MTL_SERIAL_NUMBERS_INTERFACE	
table, 5 – 42	
MTL_TRANSACTION_LOTS_INTERFACE	
table, 5 – 41	
MTL_TRANSACTIONS_INTERFACE table,	Transactions
5 – 28	importing, 8 – 2
	PA_TRANSACTION_INTERFACE table,
	8 – 7

Reader's Comment Form

Oracle Financials and Oracle Government Financials Open Interfaces Manual A36127-2

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information we use for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual? What did you like least about it?

If you find any errors or have any other suggestions for improvement, please indicate the topic, chapter, and page number below:

Please send your comments to:

Oracle Applications Documentation Manager
Oracle Corporation
500 Oracle Parkway
Redwood Shores, CA 94065 USA
Phone: (415) 506-2503 Fax: (415) 506-7800

If you would like a reply, please give your name, address, and telephone number below:

Thank you for helping us improve our documentation.



A36127



Oracle Financials®
and Oracle Government Financials™
Open Interfaces Manual

Release 10.7

Volume 2

Oracle Financials[®] and Oracle Government Financials[™] Open Interfaces Manual

**RELEASE 10.7
VOLUME 2**

February 1997

ORACLE[®]

**Your Decentralized Accounting and Management
Solution**

Oracle Financials and Oracle Government Financials Open Interfaces Manual
Release 10.7 Volume 2

The part number for this volume is A36128-2. To reorder this manual, use the set part number A22516-2.

Copyright © 1997

All rights reserved. Printed in the U.S.A.

Major Contributors: Christopher Andrews, Tracy Drinkwater, Caroline Guenther

Contributors: D. Yitzik Brenman, Steve Carter, Rick Copeland, Metin Gökçen, Anne Jaeckel, Cora Kemp, Liza Lyons, Beth Mitcham, Thornton Prayer, Maria Quinlan

This software was not developed for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It is the customer's responsibility to take all appropriate measures to ensure the safe use of such applications if the programs are used for such purposes.

This software/documentation contains proprietary information of Oracle Corporation; it is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited.

If this software/documentation is delivered to a U.S. Government Agency of the Department of Defense, then it is delivered with Restricted Rights and the following legend is applicable:

Restricted Rights Legend

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of DFARS 252.227-7013, Rights in Technical Data and Computer Software (October 1988).

Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

If this software/documentation is delivered to a U.S. Government Agency not within the Department of Defense, then it is delivered with "Restricted Rights," as defined in FAR 52.227-14, Rights in Data - General, including Alternate III (June 1987).

The information in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error-free.

Oracle Alert, Oracle Financials, Pro*C, SQL*Forms, SQL*LoaderOps, SQL*Net, SQL*Plus, SQL*Report, and SQL*ReportWriter, are registered trademarks of Oracle Corporation

Oracle Alliance Program (SM), Oracle Application Object Library, Oracle Book, CASE*Generator, Oracle7, Oracle Add-In for 1-2-3, Oracle Application Display Manager, Oracle Applications, Oracle Applications Window Manager, Oracle Assets, Oracle Business Manager, Oracle Data Browser, Oracle Data Query, Oracle General Ledger, Oracle Government Financials, Oracle Human Resources, Oracle Inventory, Oracle*Mail, Oracle Manufacturing, Oracle Order Entry, ORACLE Parallel Server, Oracle Payroll, Oracle Payables, Oracle Project Accounting, Oracle Purchasing, Oracle Receivables, PL/SQL, Pro*C, and SQL*TextRetrieval are trademarks of Oracle Corporation.



Contents

VOLUME 1

Preface	Preface i Introduction to Oracle Applications ii
Chapter 1	Open Interfaces 1 – 1 Importing Information into Oracle Financials Applications 1 – 2
Chapter 2	Oracle Assets Open Interfaces 2 – 1 Importing Asset Information into Oracle Assets Using Mass Additions 2 – 2 Integrating Oracle Assets Using the Budget Interface 2 – 38 Integrating Oracle Assets Using the Production Interface 2 – 46 Integrating Oracle Assets Using the ACE Interface 2 – 53
Chapter 3	Oracle Cash Management Open Interfaces 3 – 1
Chapter 4	Oracle General Ledger Applications Open Interfaces 4 – 1 Integrating Oracle General Ledger or Oracle Government General Ledger Using Journal Import 4 – 2 Integrating Oracle General Ledger or Oracle Government General Ledger Using Budget Upload 4 – 33

Chapter 5	Oracle Inventory Open Interfaces	5 – 1
	Open Demand Interface	5 – 2
	Open Transaction Interface	5 – 22
	Open Replenishment Interface	5 – 46
	Open Item Interface	5 – 58

Chapter 6	Oracle Order Entry Open Interfaces	6 – 1
	Ship Confirm Open Interface	6 – 2
	Integrating Oracle Order Entry with Oracle Receivables	6 – 15
	Integrating Oracle Order Entry Using OrderImport	6 – 48

Chapter 7	Oracle Payables Applications Open Interfaces	7 – 1
	Integrating Payment Reconciliation Information with Oracle Payables or Oracle Government Payables	7 – 2
	Importing Invoice Information into Oracle Payables or Oracle Government Payables	7 – 20
	Integrating Purchase Order Information with Oracle Payables or Oracle Government Payables	7 – 50

Index

VOLUME 2

Chapter 8	Oracle Project Accounting Open Interfaces	8 – 1
	Importing Transaction Data into Oracle Project Accounting	8 – 2

Chapter 9	Oracle Purchasing Applications Open Interfaces	9 – 1
	Oracle Purchasing or Oracle Government Purchasing: Open Requisitions Interface	9 – 2
	Integrating Purge Information with Oracle Payables and Oracle Purchasing Applications	9 – 23

Chapter 10

Oracle Receivables Applications Open Interfaces	10 – 1
Importing Invoice Information into Oracle Receivables	
Applications Using AutoInvoice	10 – 2
Processing Bank Receipts Using Oracle Receivables	
Applications AutoLockbox	10 – 109
Integrating Customer Information Using Oracle Receivables	
Applications Customer Interface	10 – 143
Integrating Oracle Receivables Applications Tax Information	
Using Sales Tax Rate Interface	10 – 198
Implementing the Oracle Receivables Applications	
Tax Vendor Extension	10 – 219

Index

Oracle Project Accounting Open Interfaces

This chapter contains information about the following Oracle Project Accounting open interface:

- Importing Transaction Data into Oracle Project Accounting

If you are using Oracle Project Accounting Version 3.0, please refer to the *Oracle Project Accounting Implementation Guide* for information about other Oracle Project Accounting open interfaces. If you are using Oracle Project Accounting Version 3.1, please refer to the *Oracle Project Accounting Implementation Guide* and the *Oracle Project Accounting Release 10.5 Addendum*. If you are using Oracle Projects Version 4.1, please refer to the *Oracle Projects User's Guide – Implementation Release*.

Importing Transaction Data into Oracle Project Accounting

Transaction Import is an open interface that allows you to validate and load transactions from external cost collection systems into Oracle Project Accounting. Transaction Import creates pre-approved expenditure items from transaction data entered in external cost collection systems. For example, you may wish to import transactions from the following types of external systems into Oracle Project Accounting:

- Timecard entry systems
- Expense report entry systems
- Electronic data collection systems for asset usage (computer, printer, phone, etc.)
- Payroll systems which calculate complex transactions for benefits, overtime, and other labor charges
- Fixed assets systems which calculate depreciation charged to a project

When loading transactions, Transaction Import creates expenditure groups, expenditures, and expenditure items. You can import costed, uncosted, and adjusted transactions into Oracle Project Accounting.

This section describes how to import transactions from external cost collection systems using Transaction Import. It also describes the validation Oracle Project Accounting performs for imported transactions, as well as how to resolve import exceptions.

For additional information on how Transaction Import works, refer to the following essay:



Transaction Import
(*Oracle Project Accounting Release 10.5 Addendum*)

Transaction Import Validation

Transaction Import validates your data for compatibility with Oracle Project Accounting by ensuring that the columns in the interface table reference the appropriate and active values and columns in Oracle Project Accounting.

Validating Expenditure Items

Transaction Import validates all items within an expenditure before rejecting the expenditure if one item fails validation. Transaction Import does not stop processing at the first expenditure item that fails validation, it checks all items within an expenditure.

Validating and Loading Transactions

Transaction Import validates the following attributes to ensure that your transactions contain the appropriate data for Oracle Project Accounting:

- Transaction source
- Expenditure type
- Expenditure ending date
- Employee
- Organization
- Project
- Task
- Non-labor resource and non-labor organization

Note: Read the Table Definitions section below for detailed information on additional column validation.

Transaction Import performs the following additional validation during transaction loading:

Transaction Import Validation	
Labor Items	Employee must be provided
	Employee must have an active organization and job assignment as of the expenditure item date
	Employee must have an active hourly cost rate as of the expenditure item date
Expense Report Items	Employee must be provided
	Employee must have an active organization assignment as of the expenditure item date

Table 8 – 1 (Page 1 of 2)

Transaction Import Validation	
Usage Items	A non-labor resource and a non-labor organization must be specified
	The expenditure type must be defined as a non-labor resource expenditure type
	Employee or expenditure organization must be provided
	Employee must have an active organization assignment as of the expenditure item date
	The non-labor organization must own the non-labor resource as of the expenditure item date
Additional Validation	Validates if the transaction can be charged to the project/task based on transaction controls
	If the transaction is an adjustment with a negative quantity, and the unmatched negative flag is not set to Yes, an original, approved, unreversed expenditure item matching the transaction's employee/organization, item date, expenditure type, project, task, reversing quantity, and non-labor resource and non-labor organization (for usages) must exist. Also, the matching expenditure item must have been originally loaded from the same transaction source.
	Determines the billable status of the item based on task and transaction controls

Table 8 – 1 (Page 2 of 2)

Understanding the Interface Table

You use an import utility to load transaction information into the Oracle Project Accounting interface table (PA_TRANSACTION_INTERFACE) for each transaction you want to create. When you initiate importing transactions, Transaction Import validates all required transaction data in this table. After it validates the information, Transaction Import creates transactions (expenditure items) from the information in the interface table and places the transaction information in the following expenditure tables:

- PA_EXPENDITURE_GROUPS
- PA_EXPENDITURES
- PA_EXPENDITURE_ITEMS

The interface table is organized by columns which Oracle Project Accounting uses to categorize and store specific transaction information. For example, transaction source information is stored in the column called TRANSACTION_SOURCE in PA_TRANSACTION_INTERFACE.

NOT NULL Columns

You must enter values for all NOT NULL columns in the interface table to successfully create expenditures from your imported transaction data.

NULL Columns

A NULL column is a column in the interface table that does not require a value in order for Transaction Import to successfully create transactions from the records in the table. There are two types of NULL columns:

- Some NULL columns are NULL only because they are not required for all transactions. They are required, however, depending on the type of transaction. For example, for usage items, the NON_LABOR_RESOURCE column must be populated. We mark these columns as Conditionally Required.
- Some NULL columns should never be populated because they are used by the Transaction Import program. We list these columns in the System Assigned Columns section.

Conditionally Required Columns

Oracle Project Accounting requires you to enter values in conditionally required columns only if you have entered a value in another column on which the columns are dependent. For example, if you are importing usage expenditure items, you must supply a value for the NON_LABOR_RESOURCE column; this column is not required for labor expenditure items. Any column that is not uniquely identified in the table on the following pages is conditionally required.

Optional Columns

You can use some columns in the interface table to import additional information for the transactions created by Transaction Import. Transaction Import imports the data that you load into these optional columns, provided that the information passes the validation checks that Transaction Import requires.

System Assigned Columns

Oracle Project Accounting assigns values to the columns listed below during the import process. Your import file must leave these columns blank.

- EXPENDITURE_ID
- INTERFACE_ID
- TRANSACTION_STATUS_CODE
- TRANSACTION_REJECTION_CODE

Additional Transaction Interface Tables

Oracle Project Accounting uses the PA_TRANSACTION_INTERFACE_CONTROL table to control processing of transactions by the Transaction Import program. You should not insert or update records in this table directly.

Table Descriptions

This section describes the columns in the Transaction Import interface table and indicates the columns that require values for importing transactions into Oracle Project Accounting. For a complete description of the Transaction Import interface table, please consult the *Oracle Project Accounting Technical Reference Manual*.



Additional Information: Table Definitions
(*Oracle Project Accounting Technical Reference Manual*)

Do not import your data into columns marked System Assigned in the Comments column of the following table. Columns marked OPTIONAL are for optional transaction information that you can track if you want. VARCHAR2 columns are case sensitive. You can either load the table directly from your external system, or you can fill in some values using SQL*Plus.



Attention: You must populate all NOT NULL columns before you run Transaction Import.

PA_TRANSACTION_INTERFACE

PA_TRANSACTION_INTERFACE is the table you use to import transactions from external sources into Oracle Project Accounting expenditure tables using the Transaction Import process.

Column Name	Null	Type	Comments
TRANSACTION_SOURCE	NOT NULL	VARCHAR2(30)	
BATCH_NAME	NOT NULL	VARCHAR2(10)	
EXPENDITURE_ENDING_DATE	NOT NULL	DATE	
EMPLOYEE_NUMBER		VARCHAR2(30)	
ORGANIZATION_NAME		VARCHAR2(60)	
EXPENDITURE_ITEM_DATE	NOT NULL	DATE	
PROJECT_NUMBER	NOT NULL	VARCHAR2(25)	
TASK_NUMBER	NOT NULL	VARCHAR2(25)	
EXPENDITURE_TYPE	NOT NULL	VARCHAR2(30)	
NON_LABOR_RESOURCE		VARCHAR2(20)	
NON_LABOR_RESOURCE_ORG_NAME		VARCHAR2(60)	
QUANTITY	NOT NULL	NUMBER(22,2)	
RAW_COST		NUMBER(22,2)	OPTIONAL
RAW_COST_RATE		NUMBER(22,5)	OPTIONAL
EXPENDITURE_ID		NUMBER(15)	System Assigned
INTERFACE_ID		NUMBER(15)	System Assigned
EXPENDITURE_COMMENT		VARCHAR2(240)	OPTIONAL
TRANSACTION_STATUS_CODE		VARCHAR2(1)	System Assigned, Foreign key: PA_LOOKUPS
TRANSACTION_REJECTION_CODE		VARCHAR2(30)	System Assigned, Foreign key: PA_LOOKUPS
ORIG_TRANSACTION_REFERENCE	NOT NULL	VARCHAR2(30)	
ATTRIBUTE_CATEGORY		VARCHAR2(30)	OPTIONAL

Table 8 – 2 (Page 1 of 2) PA_TRANSACTION_INTERFACE

Column Name	Null	Type	Comments
ATTRIBUTE [1-10]		VARCHAR2(150)	OPTIONAL
UNMATCHED_NEGATIVE_TXN_FLAG		VARCHAR2(1)	

Table 8 – 2 (Page 2 of 2) PA_TRANSACTION_INTERFACE

TRANSACTION_SOURCE

Enter an implementation-defined transaction source that classifies the transaction. This transaction source, along with the original transaction reference, identifies the source of transactions loaded into Oracle Project Accounting from an external system. If a transaction source is defined as *Costed*, then a raw cost amount must exist for the transaction.

Validation: The transaction source you enter must be a valid transaction source. You can obtain a list of valid transaction sources from PA_TRANSACTION_SOURCES.TRANSACTION_SOURCE.

Destination: PA_EXPENDITURE_GROUPS.TRANSACTION_SOURCE and PA_EXPENDITURE_ITEMS.TRANSACTION_SOURCE. The transaction source information is denormalized for performance optimization.

BATCH_NAME

Enter a user-defined batch name that identifies the transaction batch. A transaction batch is a group of expenditures loaded into the interface table; all transactions in this group must have the same transaction source. The batch name is used to derive part of the expenditure group name in Oracle Project Accounting expenditure tables after import.

Validation: None

Destination: PA_EXPENDITURE_GROUPS.EXPENDITURE_GROUP

EXPENDITURE_ENDING_DATE

Enter the date of the last day of the expenditure week for this transaction. All transactions in an expenditure must be on or before the expenditure ending date. In addition, all timecard items must be

within the expenditure week date range. The maximum expenditure ending date of all expenditure items processed in a group becomes the expenditure group ending date.

Validation: Valid week ending date based on the expenditure cycle start day defined in Implementation Options.

Destination: PA_EXPENDITURES.EXPENDITURE_ENDING_DATE

EMPLOYEE_NUMBER

Enter the number of the employee who incurred the charge for this transaction. This column must be populated for labor and expense report items, but is optional for usages.

Validation: Must be a valid employee number in PER_PEOPLE_F.EMPLOYEE_NUMBER

Destination: PA_EXPENDITURES.INCURRED_BY_PERSON_ID

ORGANIZATION_NAME

Enter the name of the organization that incurred the charge for this transaction. If employee number is provided, then this column can be null, in which case Transaction Import derives this value from the employee organization. If you provide both an employee and an organization, Oracle Project Accounting uses the employee information to derive the organization. Transaction Import uses the last employee assignment in the expenditure period to derive the employee organization.

Validation: Must be a valid organization in PER_ORGANIZATION_UNITS.NAME

Destination: PA_EXPENDITURES.INCURRED_BY_ORGANIZATION_ID

EXPENDITURE_ITEM_DATE

Enter the date on which this transaction occurred.

Validation: The expenditure item date must be on or before the expenditure ending date. Also, the expenditure item date of timecard items must fall within the expenditure week as defined by the expenditure ending date.

Destination: PA_EXPENDITURE_ITEMS.EXPENDITURE_ITEM_DATE

PROJECT_NUMBER

Enter the number of the project this transaction is charged to.

Validation: Must be a valid project number in PA_PROJECTS.SEGMENT1

Destination: None

TASK_NUMBER

Enter the number of the task this transaction is charged to.

Validation: Must be a valid task number in PA_TASKS.TASK_NUMBER for the project number specified.

Destination: PA_EXPENDITURE_ITEMS.TASK_ID

EXPENDITURE_TYPE

Enter the expenditure type that classifies the type of charge for this transaction.

Validation: This expenditure type must be a valid expenditure type in PA_EXPENDITURE_TYPES.EXPENDITURE_TYPE. Also, the system linkage of the expenditure type must match the system linkage of the transaction source. You cannot import expenditure items with a system linkage of Overtime or Vendor Invoices.

Destination: PA_EXPENDITURE_ITEMS.EXPENDITURE_TYPE

NON_LABOR_RESOURCE

Enter the non-labor resource utilized for this transaction. This column is populated only for usage items.

Validation: This non-labor resource must be a valid non-labor resource in PA_NON_LABOR_RESOURCES.NON_LABOR_RESOURCE and must be a resource classified by the specified expenditure type.

Destination: PA_EXPENDITURE_ITEMS.NON_LABOR_RESOURCE

NON_LABOR_RESOURCE_ORG_NAME

Enter the name of the organization owning the non-labor resource utilized for the transaction. This column is populated only for usage items.

Validation: Must be a valid organization in PER_ORGANIZATION_UNITS.NAME. Must also be a valid non-labor resource owning organization in PA_NON_LABOR_RESOURCE_ORGS.ORGANIZATION_ID for the specified non-labor resource.

Destination: PA_EXPENDITURE_ITEMS.ORGANIZATION_ID

QUANTITY

Enter the number of units for the transaction.

Validation: None

Destination: PA_EXPENDITURE_ITEMS.QUANTITY

RAW_COST

Enter the total raw cost of the transaction.

Validation: If the transaction source is defined as *Costed*, a raw cost amount must exist. If transaction source is not *Costed*, this column value is ignored.

Destination: PA_EXPENDITURE_ITEMS.RAW_COST

RAW_COST_RATE

Enter the raw cost rate used to cost the transaction. Oracle Project Accounting uses this information for reporting purposes only.

Validation: None

Destination: PA_EXPENDITURE_ITEMS.RAW_COST_RATE

EXPENDITURE_ID

This column is populated by a system-defined value to identify the transactions grouped into an expenditure.

Validation: This column is system assigned.
Destination: PA_EXPENDITURES.EXPENDITURE_ID

INTERFACE_ID

This column is populated by a system-defined value to identify transactions processed by a given concurrent request.

Validation: This column is system assigned.
Destination: None

EXPENDITURE_COMMENT

Enter the description that you want to assign to the expenditure item created from this transaction.

Validation: None
Destination: PA_EXPENDITURE_COMMENTS.EXPENDITURE_COMMENT

TRANSACTION_STATUS_CODE

This column is populated by a system-defined status of the transaction as it is loaded into Oracle Project Accounting.

Validation: This column is system assigned. Lookup codes for this column are stored in the PA_LOOKUPS table under the lookup type of TRANSACTION STATUS.
Destination: None

TRANSACTION_REJECTION_CODE

This column is populated by a system-defined reason why the transaction was rejected by the Transaction Import program.

Validation: This column is system assigned. Lookup codes for this column are stored in the PA_LOOKUPS table under the lookup type of TRANSACTION REJECTION REASON.
Destination: None

ORIG_TRANSACTION_REFERENCE

Enter a unique reference to the original item loaded into Oracle Project Accounting from an external system via the Transaction Import program. This reference, along with the transaction source, identifies the source of the original transaction loaded into Oracle Project Accounting from an external system.

Validation: An expenditure item must not already exist with the same identifier values as the transaction.

Destination: PA_EXPENDITURE_ITEMS.ORIG_TRANSACTION_REFERENCE

ATTRIBUTE_CATEGORY

Enter the descriptive flexfield category for the descriptive flexfield information you want to import.

Validation: None

Destination: PA_EXPENDITURE_ITEMS.ATTRIBUTE_CATEGORY

ATTRIBUTE [1-10]

Enter the descriptive flexfield information that you want to import for a transaction (expenditure item). The structure of the information you enter in these columns (datatypes, value sets) should match the structure of the descriptive flexfield segments you have defined for your transaction or you will experience validation problems when you try to access the information in the expenditure entry forms.

Validation: None

Destination: PA_EXPENDITURE_ITEMS.ATTRIBUTE [1-10]

UNMATCHED_NEGATIVE_TXN_FLAG

Enter Y or N so that Transaction Import can identify summary-level adjustments (negative amounts) for which there is no single matching item to adjust. If this column is set to Yes, Transaction Import will bypass the matching validation logic that is usually executed for adjustments (negative transactions).

Validation: None

Destination: None

Implementation Data

Define Transaction Sources

Use the **Define Transaction Sources** form to define the source of transactions for Transaction Import. You can define an unlimited number of transaction sources; Oracle Project Accounting does not predefine any transaction sources. For each transaction source, you specify whether it is purgeable; Transaction Import automatically purges transactions that are purgeable from the interface table after they are successfully imported into Oracle Project Accounting.



Defining Transaction Sources

(Oracle Project Accounting Release 10.5 Addendum)

You use your import utility to enter this transaction source in the TRANSACTION_SOURCE column of the PA_TRANSACTION_INTERFACE table. You then select the name in the **Submit Processes** form when you want to import transactions from this source.

Populating the Interface Table

Transaction Import depends on transaction data from your external system to create corresponding transactions in Oracle Project Accounting. Before you can import transactions, you must first write a custom feeder program that transfers transaction data from your original system into the Transaction Import interface table. Your feeder program must convert data from your original system into a standard data format that Transaction Import can read. Transaction Import can then convert your imported data into transactions.

Writing a Feeder Program

The type of environment from which you want to transfer your data determines the type of feeder program you need to write. For example, you can use SQL*Loader, SQL*Report, PL/SQL, or Pro*C to write a feeder program to transfer transaction data from a non-Oracle system. Or, you can write a conversion program to transfer historical data from your previous cost collection system.

Ensure that your transaction flat file has the appropriate information to populate PA_TRANSACTION_INTERFACE as indicated in the preceding table descriptions. If a value is not required for a column, you may leave the column empty.

Selecting an Import Utility

SQL*Loader and SQL*Report are powerful and easy-to-use tools that should be able to accommodate all of your import needs. However, depending on the complexity of your import program, you may also want to use Oracle's Pro* language products such as Pro*C, Pro*Cobol and Pro*Fortran to write the program.

Your import utility file must populate PA_TRANSACTION_INTERFACE as indicated in the previous table descriptions. Also, you should code your file to populate the TRANSACTION_SOURCE column in PA_TRANSACTION_INTERFACE with the Transaction Source code exactly as you defined it in the **Define Transaction Sources** form.

You must provide any information that the Oracle Project Accounting interface table requires that your external system does not provide. For example, if your external timecard system does not provide expenditure types, you must create at least one expenditure type and specify it in your control file.

Importing Transactions

Complete the following steps after you populate the interface table to import external transactions into Oracle Project Accounting:

Spot Check the Interface Table

Check several records throughout the interface table and compare them to the corresponding records in the original transaction data file. Look for missing or invalid data. This step ensures that your data was imported into the correct columns and that all columns were imported.

Run Transaction Import

Run Transaction Import to validate and load transactions into Oracle Project Accounting as pre-approved expenditure items. Read the Transaction Import process essay for more information.



Transaction Import
(Oracle Project Accounting Release 10.5 Addendum)

Viewing Successful Transactions

The Transaction Import report displays a summary of the expenditure items and expenditures successfully imported into Oracle Project Accounting by the Transaction Import process. This report lists the name of each expenditure group created during the import process, the expenditure group ending date, and the total number of expenditures created in the expenditure group. This report also includes a total count of the expenditure groups and expenditure created during the import process.

For detailed information on successfully imported expenditure items, use the following information as parameters for the AUD: Pre-Approved Expenditure Entry Audit Report. The information for these parameters is displayed in the Transaction Import output report.

- Expenditure group
- Employee name that corresponds to the userid of the person that submitted Transaction Import



Transaction Import
(*Oracle Project Accounting Release 10.5 Addendum*)



Pre-Approved Expenditure Entry Audit Report
(*Oracle Project Accounting Reference Manual*)

For information about viewing successful transactions in Oracle Project Accounting, refer to the following essay:



Transaction Import
(*Oracle Project Accounting Release 10.5 Addendum*)

Resolving Import Exceptions

A rejected transaction must be corrected before it can be loaded into Oracle Project Accounting. You can correct transaction data in the interface table or in your external feeder system and reload the data. This section describes how to correct rejected data, and also describes reports you can use to help resolve exceptions.



Attention: If you correct exceptions in your external system, you must delete the rejected rows from the interface table before reloading the corrected transactions.

Rejection Reasons

Transaction Import may reject importing transactions for one or more of the following reasons:

DUPLICATE_ITEM	A transaction with the same transaction source and original transaction reference has already been imported into Oracle Project Accounting.
INVALID_END_DATE	The value for the expenditure ending date is not a valid week ending date
INVALID_PROJECT	No project exists with the project number specified.
ITEM_NOT_IN_WEEK	The expenditure item date for a timecard item does not fall within the timecard expenditure week.
PA_EXP_TASK_TC	The transaction violates an expenditure transaction control at the task level.
PA_EX_PROJECT_CLOSED	The transaction is being charged to a closed project; you cannot charge items to a closed project.

Viewing Rejected Transactions

Transaction records that fail the validation process remain in the interface table. There are two methods you can use to view failed transactions:

- Use SQL*Plus
- Review an Oracle Project Accounting report

SQL*Plus

You can use SQL*Plus to identify the records that have failed by selecting those rows with a TRANSACTION_STATUS_CODE of *R* and selecting the rejection reason for each rejected record from the TRANSACTION_REJECTION_CODE column.

Report

The Transaction Import Exception Report shows you all of the transactions that were rejected during the Transaction Import process. For each rejected transaction, this report displays the key field values of the transaction in the interface table. It also displays the rejection reason code that identifies the cause of the transaction's rejection.



Transaction Import Exception Report
(Oracle Project Accounting Release 10.5 Addendum)

Correcting Rejected Transactions

If you need to make changes to the source information because of invalid data, you need to delete the rejected rows in the interface table, correct the failed transactions in the feeder system, and reload them from the feeder system. You can also update the transaction in the interface table using SQL*Plus. When you resubmit updated transactions for processing, all validation is performed again.



Attention: If any one expenditure item in an expenditure fails validation, Oracle Project Accounting rejects the entire expenditure and updates each expenditure item in the expenditure with a status of R (Rejected). However, only the expenditure item that was rejected appears on the exception report, other expenditure items attached to the expenditure being rejected do not appear on the report. Also, only the transaction having invalid data will have a rejection reason specified. Figure 8 – 1 demonstrates these concepts.

Figure 8 – 1

Before Transaction Import:			
	Status	Reason	Expenditure ID
Transaction 1			
Transaction 2			
Transaction 3			

After Transaction Import:			
	Status	Reason	Expenditure ID
Transaction 1 *	R	INVALID_PROJECT	1009
Transaction 2	R		1009
Transaction 3	R		1009

* This is the only item that is displayed on the exceptions report



Suggestion: You can locate all of the related rejected transactions within an expenditure by using a SQL*Plus select statement on the EXPENDITURE_ID column using the expenditure id of the rejected item; then update the TRANSACTION_STATUS_CODE column to remove the R status.

Example of Steps You Take to Fix a Rejected Transaction

Let's walk through an example of the steps you take to correct a rejected transaction using the rejected transaction in Figure 8 – 1 as our sample data.

1. Correct the invalid data for Transaction 1

Transaction 1 was rejected because the project you are charging the expenditure item to is invalid. Using SQL*Plus, you update the project number of the transaction to a valid project number.

2. Update all of the transactions in the expenditure

Now that the source of the rejected item has been corrected, you need to update all of the transactions within the expenditure. You can identify all of the rejected transactions within the rejected expenditure by the Expenditure ID, which in this case is 1009. You can also identify the status code of the rejected expenditure items by the value in the TRANSACTION_STATUS_CODE column, which in this case is R. Using SQL*Plus, you update the TRANSACTION_STATUS_CODE to a status of pending by setting the column to NULL for all expenditure IDs having a value of 1009:

```
update PA_TRANSACTION_INTERFACE
set TRANSACTION_STATUS_CODE = NULL
where EXPENDITURE_ID = 1009;
```

3. Run Transaction Import

Now that you have corrected the rejected expenditure item and updated the status of all expenditure items within the rejected expenditure, you can run Transaction Import to successfully import the updated transactions.

Oracle Purchasing Applications Open Interfaces

This chapter contains information about the following Oracle Purchasing Applications open interfaces:

- Open Requisitions Interface
- Integrating Purge Information with Oracle Payables and Oracle Purchasing Applications

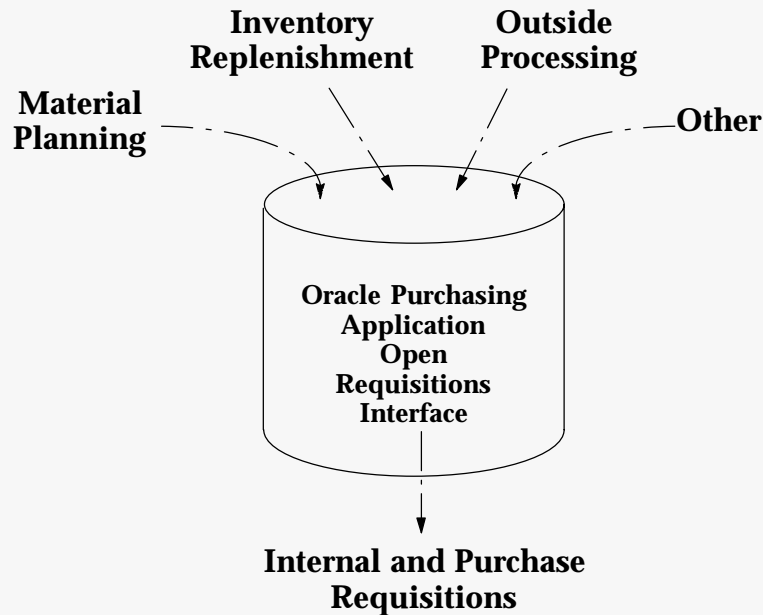
Oracle Purchasing or Oracle Government Purchasing: Open Requisitions Interface

You can automatically import requisitions from other Oracle Applications or your existing non-Oracle systems using the Open Requisitions Interface. This interface lets you integrate your Oracle Purchasing application quickly with new or existing applications such as material requirements planning, inventory management, and production control systems. Your Oracle Purchasing application automatically validates your data and imports your requisitions. You can import requisitions as often as you want. Then, you can review these requisitions, approve or reserve funds for them if necessary, and place them on purchase orders or internal sales orders.

The purpose of this essay is to explain how to use the Open Requisitions Interface so that you can integrate other applications with your Oracle Purchasing application.

Functional Overview

Figure 9 – 1



The diagram above shows the inputs and outputs that comprise the interface process.

You must write the program that inserts a single row into the PO_REQUISITIONS_INTERFACE table for each requisition line that you want to import. Then you use the Run Reports form to launch the Requisition Import program for any set of rows. You identify the set of rows you want to import by setting the INTERFACE_SOURCE_CODE and BATCH_ID columns appropriately in the interface table. You then pass these values as parameters to the Requisition Import program. If you do not specify any values for these parameters, the program imports all the requisition lines in the interface table. You also specify the requisition grouping and numbering criteria as parameters to the Requisition Import program.

The Requisition Import program operates in three phases. In the first phase, the program validates your data and derives or defaults additional information. The program generates an error message for every validation that fails and creates a row in the

PO_INTERFACE_ERRORS table with detailed information about each error.

In the second phase, the program groups and numbers the validated requisition lines according to the following criteria. If you specify a value in the REQ_NUMBER_SEGMENT1 column of the interface table, all lines with the same value for this column are grouped together under a requisition header. If you provide a value in the GROUP_CODE column, all lines with the same value in this column are grouped together under a requisition header. If you do not provide values in either of these columns, the Requisition Import program uses the Group By parameter to group lines together. If you do not provide a value for this parameter, the program uses the Default Group By that you set up to group requisition lines. You can group requisition lines in one of the following ways that the Requisition Import program supports:

- by BUYER
- by CATEGORY
- by LOCATION
- by VENDOR
- by ITEM
- by ALL (all requisition lines grouped under one header)

If you provide a value in the REQ_NUMBER_SEGMENT1 column, this value becomes the requisition number. If not, the Requisition Import program uses either the Last Requisition Number parameter if specified or the next unique number stored in the PO_UNIQUE_IDENTIFIER_CONTROL table, adds 1 to this number, and starts numbering requisitions. If any of the requisition numbers generated already exists, the program loops until it finds a unique number. For every line that is successfully imported, a default distribution is created with the account information that you specify. Requisition supply is also created for every approved requisition that is successfully imported.

In the third phase, the program deletes all the successfully processed rows and creates a report which lists the number of interface records that were successfully imported and the number that were not imported. This report can be viewed by picking the Report option from the View field of the View Concurrent Requests form for the Requisition Import concurrent request ID.

You can launch the Requisition Import Exceptions Report to view the rows that were not imported by the Requisition Import program along with the failure reason(s) for each row.

You can import approved or unapproved requisitions using the Open Requisitions Interface. If you are using requisition encumbrance, approved requisitions that you import automatically become pre-approved, and you can then reserve funds for them using the Approve Documents form.

Setting Up the Requisitions Interface

You must complete the following setup steps of your Oracle Purchasing application setup to use the Open Requisitions Interface. You must define a ReqImport Group By method in the Default Options zone of the Define Purchasing Options form. For internally sourced requisitions you must associate a customer with your deliver-to location using the Define Location Associations form.

All processing is initiated via Standard Report Submission(SRS) using the Run Reports form. The concurrent manager manages all processing, and as such it must be set up and running.

Inserting into the Requisitions Interface Table

You load requisition lines from your source system or form into the requisitions interface table. You insert one row for each requisition line that you want to import. You must provide values for all columns that are required. You may also have to provide values for columns that are conditionally required.

Requisitions Interface Table Description

The following graphic describes the requisitions interface table:

PO_REQUISITIONS_INTERFACE Column Name	Type	Required	Derived	Optional
TRANSACTION_ID	Number		Y	
PROCESS_FLAG	Varchar2		Y	
REQUEST_ID	Number		Y	
PROGRAM_ID	Number		Y	
PROGRAM_APPLICATION_ID	Number		Y	
PROGRAM_UPDATE_DATE	Date		Y	
LAST_UPDATED_BY	Number		Y	
LAST_UPDATE_DATE	Date		Y	
LAST_UPDATE_LOGIN	Number		Y	
CREATION_DATE	Date			Y
CREATED_BY	Number			Y
INTERFACE_SOURCE_CODE	Varchar2			Y
INTERFACE_SOURCE_LINE_ID	Number			Y
BATCH_ID	Number			Y
GROUP_CODE	Varchar2			Y
SOURCE_TYPE_CODE	Varchar2	Y	C	

PO_REQUISITIONS_INTERFACE				
Column Name	Type	Required	Derived	Optional
REQUISITION_TYPE	Varchar2		Y	
DESTINATION_TYPE_CODE	Varchar2	Y		
AUTHORIZATION_STATUS	Varchar2	Y		
PREPARER_ID	Number	Y	C	
PREPARER_NAME	Varchar2			Y
APPROVER_ID	Number		Y	
APPROVER_NAME	Varchar2			Y
APPROVAL_PATH_ID	Number			Y
REQUISITION_HEADER_ID	Number		Y	
REQUISITION_LINE_ID	Number		Y	
REQ_DISTRIBUTION_ID	Number		Y	
REQ_NUMBER_SEGMENT1	Number			Y
REQ_NUMBER_SEGMENT2	Number			Y
REQ_NUMBER_SEGMENT3	Number			Y
REQ_NUMBER_SEGMENT4	Number			Y
REQ_NUMBER_SEGMENT5	Number			Y
HEADER_DESCRIPTION	Varchar2			Y
HEADER_ATTRIBUTE_CATEGORY	Varchar2			Y
HEADER_ATTRIBUTE1	Varchar2			Y
HEADER_ATTRIBUTE2	Varchar2			Y
HEADER_ATTRIBUTE3	Varchar2			Y
HEADER_ATTRIBUTE4	Varchar2			Y
HEADER_ATTRIBUTE5	Varchar2			Y

PO_REQUISITIONS_INTERFACE				
Column Name	Type	Required	Derived	Optional
HEADER_ATTRIBUTE6	Varchar2			Y
HEADER_ATTRIBUTE7	Varchar2			Y
HEADER_ATTRIBUTE8	Varchar2			Y
HEADER_ATTRIBUTE9	Varchar2			Y
HEADER_ATTRIBUTE10	Varchar2			Y
HEADER_ATTRIBUTE11	Varchar2			Y
HEADER_ATTRIBUTE12	Varchar2			Y
HEADER_ATTRIBUTE13	Varchar2			Y
HEADER_ATTRIBUTE14	Varchar2			Y
HEADER_ATTRIBUTE15	Varchar2			Y
URGENT_FLAG	Varchar2			Y
RFQ_REQUIRED_FLAG	Varchar2			Y
JUSTIFICATION	Varchar2			Y
NOTE_TO_BUYER	Varchar2			Y
NOTE_TO_RECEIVER	Varchar2			Y
NOTE_TO_APPROVER	Varchar2			Y
ITEM_ID	Number	C	C	
ITEM_SEGMENT1	Varchar2			Y
ITEM_SEGMENT2	Varchar2			Y
ITEM_SEGMENT3	Varchar2			Y
ITEM_SEGMENT4	Varchar2			Y
ITEM_SEGMENT5	Varchar2			Y
ITEM_SEGMENT6	Varchar2			Y

PO_REQUISITIONS_INTERFACE				
Column Name	Type	Required	Derived	Optional
ITEM_SEGMENT7	Varchar2			Y
ITEM_SEGMENT8	Varchar2			Y
ITEM_SEGMENT9	Varchar2			Y
ITEM_SEGMENT10	Varchar2			Y
ITEM_SEGMENT11	Varchar2			Y
ITEM_SEGMENT12	Varchar2			Y
ITEM_SEGMENT13	Varchar2			Y
ITEM_SEGMENT14	Varchar2			Y
ITEM_SEGMENT15	Varchar2			Y
ITEM_SEGMENT16	Varchar2			Y
ITEM_SEGMENT17	Varchar2			Y
ITEM_SEGMENT18	Varchar2			Y
ITEM_SEGMENT19	Varchar2			Y
ITEM_SEGMENT20	Varchar2			Y
ITEM_DESCRIPTION	Varchar2		Y	
ITEM_REVISION	Varchar2			Y
CATEGORY_ID	Number	C	C	
CATEGORY_SEGMENT1	Varchar2			Y
CATEGORY_SEGMENT2	Varchar2			Y
CATEGORY_SEGMENT3	Varchar2			Y
CATEGORY_SEGMENT4	Varchar2			Y
CATEGORY_SEGMENT5	Varchar2			Y
CATEGORY_SEGMENT6	Varchar2			Y
CATEGORY_SEGMENT7	Varchar2			Y

PO_REQUISITIONS_INTERFACE				
Column Name	Type	Required	Derived	Optional
CATEGORY_SEGMENT8	Varchar2			Y
CATEGORY_SEGMENT9	Varchar2			Y
CATEGORY_SEGMENT10	Varchar2			Y
CATEGORY_SEGMENT11	Varchar2			Y
CATEGORY_SEGMENT12	Varchar2			Y
CATEGORY_SEGMENT13	Varchar2			Y
CATEGORY_SEGMENT14	Varchar2			Y
CATEGORY_SEGMENT15	Varchar2			Y
CATEGORY_SEGMENT16	Varchar2			Y
CATEGORY_SEGMENT17	Varchar2			Y
CATEGORY_SEGMENT18	Varchar2			Y
CATEGORY_SEGMENT19	Varchar2			Y
CATEGORY_SEGMENT20	Varchar2			Y
QUANTITY	Number	Y		
UNIT_PRICE	Number		Y	
CHARGE_ACCOUNT_ID	Number	Y	C	
CHARGE_ACCOUNT_SEGMENT1	Varchar2			Y
CHARGE_ACCOUNT_SEGMENT2	Varchar2			Y
CHARGE_ACCOUNT_SEGMENT3	Varchar2			Y
CHARGE_ACCOUNT_SEGMENT4	Varchar2			Y
CHARGE_ACCOUNT_SEGMENT5	Varchar2			Y
CHARGE_ACCOUNT_SEGMENT6	Varchar2			Y
CHARGE_ACCOUNT_SEGMENT7	Varchar2			Y
CHARGE_ACCOUNT_SEGMENT8	Varchar2			Y
CHARGE_ACCOUNT_SEGMENT9	Varchar2			Y

PO_REQUISITIONS_INTERFACE				
Column Name	Type	Required	Derived	Optional
CHARGE_ACCOUNT_SEGMENT10	Varchar2			Y
CHARGE_ACCOUNT_SEGMENT11	Varchar2			Y
CHARGE_ACCOUNT_SEGMENT12	Varchar2			Y
CHARGE_ACCOUNT_SEGMENT13	Varchar2			Y
CHARGE_ACCOUNT_SEGMENT14	Varchar2			Y
CHARGE_ACCOUNT_SEGMENT15	Varchar2			Y
CHARGE_ACCOUNT_SEGMENT16	Varchar2			Y
CHARGE_ACCOUNT_SEGMENT17	Varchar2			Y
CHARGE_ACCOUNT_SEGMENT18	Varchar2			Y
CHARGE_ACCOUNT_SEGMENT19	Varchar2			Y
CHARGE_ACCOUNT_SEGMENT20	Varchar2			Y
ACCRUAL_ACCOUNT_ID	Number		Y	
VARIANCE_ACCOUNT_ID	Number		Y	
BUDGET_ACCOUNT_ID	Number		Y	
UNIT_OF_MEASURE	Varchar2	C	C	
UOM_CODE	Varchar2			Y
LINE_TYPE_ID	Number		Y	
LINE_TYPE	Varchar2			Y
UN_NUMBER_ID	Number		C	
UN_NUMBER	Varchar2			Y
HAZARD_CLASS_ID	Number		C	
HAZARD_CLASS	Varchar2			Y
REFERENCE_NUM	Varchar2			Y
SOURCE_ORGANIZATION_ID	Number		C	
SOURCE_ORGANIZATION_CODE	Varchar2			Y
SOURCE_SUBINVENTORY	Varchar2			Y

PO_REQUISITIONS_INTERFACE				
Column Name	Type	Required	Derived	Optional
DESTINATION_ORGANIZATION_ID	Number	Y	C	
DESTINATION_ORGANIZATION_CODE	Number			Y
DESTINATION_SUBINVENTORY	Varchar2			Y
DELIVER_TO_LOCATION_ID	Number	Y	C	
DELIVER_TO_LOCATION_CODE	Varchar2			Y
DELIVER_TO_REQUESTOR_ID	Number	Y	C	
DELIVER_TO_REQUESTOR_NAME	Varchar2			Y
AUTOSOURCE_FLAG	Varchar2			Y
AUTOSOURCE_DOC_HEADER_ID	Number		C	
AUTOSOURCE_DOC_LINE_NUM	Number		C	
DOCUMENT_TYPE_CODE	Varchar2		C	
SUGGESTED_BUYER_ID	Number		C	
SUGGESTED_BUYER_NAME	Varchar2			Y
SUGGESTED_VENDOR_ID	Number		C	
SUGGESTED_VENDOR_NAME	Varchar2			Y
SUGGESTED_VENDOR_SITE_ID	Number		C	
SUGGESTED_VENDOR_SITE	Varchar2			Y
SUGGESTED_VENDOR_CONTACT_ID	Number		C	
SUGGESTED_VENDOR_CONTACT	Varchar2		C	
SUGGESTED_VENDOR_PHONE	Varchar2		C	
SUGGESTED_VENDOR_ITEM_NUM	Varchar2			Y
LINE_ATTRIBUTE_CATEGORY	Varchar2			Y
LINE_ATTRIBUTE1	Varchar2			Y

PO_REQUISITIONS_INTERFACE				
Column Name	Type	Required	Derived	Optional
LINE_ATTRIBUTE2	Varchar2			Y
LINE_ATTRIBUTE3	Varchar2			Y
LINE_ATTRIBUTE4	Varchar2			Y
LINE_ATTRIBUTE5	Varchar2			Y
LINE_ATTRIBUTE6	Varchar2			Y
LINE_ATTRIBUTE7	Varchar2			Y
LINE_ATTRIBUTE8	Varchar2			Y
LINE_ATTRIBUTE9	Varchar2			Y
LINE_ATTRIBUTE10	Varchar2			Y
LINE_ATTRIBUTE11	Varchar2			Y
LINE_ATTRIBUTE12	Varchar2			Y
LINE_ATTRIBUTE13	Varchar2			Y
LINE_ATTRIBUTE14	Varchar2			Y
LINE_ATTRIBUTE15	Varchar2			Y
NEED_BY_DATE	Date	C		
NOTE1_ID	Number		C	
NOTE2_ID	Number		C	
NOTE3_ID	Number		C	
NOTE4_ID	Number		C	
NOTE5_ID	Number		C	
NOTE6_ID	Number		C	
NOTE7_ID	Number		C	
NOTE8_ID	Number		C	
NOTE9_ID	Number		C	
NOTE10_ID	Number		C	

PO_REQUISITIONS_INTERFACE				
Column Name	Type	Required	Derived	Optional
NOTE1_TITLE	Varchar2			Y
NOTE2_TITLE	Varchar2			Y
NOTE3_TITLE	Varchar2			Y
NOTE4_TITLE	Varchar2			Y
NOTE5_TITLE	Varchar2			Y
NOTE6_TITLE	Varchar2			Y
NOTE7_TITLE	Varchar2			Y
NOTE8_TITLE	Varchar2			Y
NOTE9_TITLE	Varchar2			Y
NOTE10_TITLE	Varchar2			Y
DIST_ATTRIBUTE_CATEGORY	Varchar2			Y
DISTRIBUTION_ATTRIBUTE1	Varchar2			Y
DISTRIBUTION_ATTRIBUTE2	Varchar2			Y
DISTRIBUTION_ATTRIBUTE3	Varchar2			Y
DISTRIBUTION_ATTRIBUTE4	Varchar2			Y
DISTRIBUTION_ATTRIBUTE5	Varchar2			Y
DISTRIBUTION_ATTRIBUTE6	Varchar2			Y
DISTRIBUTION_ATTRIBUTE7	Varchar2			Y
DISTRIBUTION_ATTRIBUTE8	Varchar2			Y
DISTRIBUTION_ATTRIBUTE9	Varchar2			Y
DISTRIBUTION_ATTRIBUTE10	Varchar2			Y
DISTRIBUTION_ATTRIBUTE11	Varchar2			Y
DISTRIBUTION_ATTRIBUTE12	Varchar2			Y
DISTRIBUTION_ATTRIBUTE13	Varchar2			Y
DISTRIBUTION_ATTRIBUTE14	Varchar2			Y
DISTRIBUTION_ATTRIBUTE15	Varchar2			Y

PO_REQUISITIONS_INTERFACE				
Column Name	Type	Required	Derived	Optional
GOVERNMENT_CONTEXT	Varchar2			Y
GL_DATE	Date		C	
USSGL_TRANSACTION_CODE	Varchar2			Y
PREVENT_ENCUMBRANCE_FLAG	Varchar2		Y	
CURRENCY_CODE	Varchar2			Y
CURRENCY_UNIT_PRICE	Number		C	
RATE	Number		C	
RATE_DATE	Date	C		
RATE_TYPE	Varchar2	C		
WIP_ENTITY_ID	Number	C		
WIP_LINE_ID	Number			Y
WIP_OPERATION_SEQ_NUM	Number			Y
WIP_RESOURCE_SEQ_NUM	Number			Y
WIP_REPETITIVE_SCHEDULE_ID	Number	C		
BOM_RESOURCE_ID	Number	C		
EXPENDITURE_ORGANIZATION_ID	Number	C		
EXPENDITURE_TYPE	Varchar2	C		
PROJECT_ACCOUNTING_CONTEXT	Varchar2			Y
PROJECT_ID	Number	C	C	
PROJECT_NUM	Varchar2			Y
TASK_ID	Number	C	C	
TASK_NUM	Varchar2			Y
EXPENDITURE_ITEM_DATE	Date			Y
TRANSACTION_REASON_CODE	Varchar2			Y

Table 9 – 1 Oracle Purchasing Application Open Requisitions Interface Table

LEGEND:

Y – Required

C – Conditionally required

Required Data

You must always enter values for the following required columns when you load rows into the PO_REQUISITIONS_INTERFACE table:

- DESTINATION_TYPE_CODE
- QUANTITY
- AUTHORIZATION_STATUS
- PREPARER_ID or PREPARER_NAME
- CHARGE_ACCOUNT_ID or charge account segment values
- DESTINATION_ORGANIZATION_ID or
DESTINATION_ORGANIZATION_CODE
- DELIVER_TO_LOCATION_ID or
DELIVER_TO_LOCATION_CODE
- DELIVER_TO_REQUESTOR_ID or
DELIVER_TO_REQUESTOR_NAME

Additionally, you may have to enter values for the following conditionally required columns.

You must provide a SOURCE_TYPE_CODE if the DESTINATION_TYPE_CODE is 'EXPENSE' or 'SHOP FLOOR'. You must provide an ITEM_ID or item segment values if the SOURCE_TYPE_CODE or DESTINATION_TYPE_CODE is 'INVENTORY'. For one-time items and amount-based line types, you must provide a CATEGORY_ID or category segment values. You must additionally provide a UNIT_OF_MEASURE or UOM_CODE for one-time items. For MRP or Inventory planned items, you must also provide a NEED_BY_DATE.

You must provide the RATE_DATE and RATE_TYPE if you provide a value in the CURRENCY_CODE column.

If you are using Oracle Work in Process and the DESTINATION_TYPE_CODE is 'SHOP FLOOR', you must provide values for the following columns:

- WIP_ENTITY_ID
- BOM_RESOURCE_ID
- WIP_REPETITIVE_SCHEDULE_ID, if the entity is a repetitive schedule

If you are using Oracle Project Accounting and the PROJECT_ACCOUNTING_CONTEXT is 'Y', you must enter the relevant project accounting information in the following columns:

- PROJECT_ID
- TASK_ID
- EXPENDITURE_TYPE
- EXPENDITURE_ORGANIZATION_ID.

Derived Data

The Requisition Import program derives or defaults the columns identified as derived using logic similar to that used by the Enter Requisitions form. Your Oracle Purchasing application never overrides information that you provide in derived columns. (Vendor sourcing is an exception to this rule). Column pairs like APPROVER_ID / APPROVER_NAME and NOTE_ID / NOTE_TITLE allow you to enter the user displayed value in the interface table and the program derives the associated unique identifier. If there is a conflict between the two values, the identifier overrides the user displayed value.

You can provide the segment values for the item, category, and charge account. The Requisition Import program derives the ITEM_ID, CATEGORY_ID, and CHARGE_ACCOUNT_ID. The ACCRUAL_ACCOUNT_ID, BUDGET_ACCOUNT_ID, and VARIANCE_ACCOUNT_ID are derived based on the DESTINATION_TYPE_CODE.

For interface lines with a DESTINATION_TYPE_CODE of 'INVENTORY', the program derives the SOURCE_TYPE_CODE. The REQUISITION_TYPE is derived from the SOURCE_TYPE_CODE.

The Requisition Import program automatically derives sourcing information for both your inventory and purchase requisition lines if you set the AUTOSOURCE_FLAG to 'Y' and set up the sourcing rules for the item. For inventory sourced requisition lines, the program derives the following columns:

- SOURCE_ORGANIZATION_ID
- SOURCE_SUBINVENTORY

For vendor sourced requisition lines, the program derives the following columns:

- SUGGESTED_VENDOR_ID
- SUGGESTED_VENDOR_SITE_ID

- SUGGESTED_VENDOR_CONTACT_ID
- SUGGESTED_BUYER_ID
- AUTOSOURCE_DOCUMENT_HEADER_ID
- AUTOSOURCE_DOCUMENT_LINE_NUM
- DOCUMENT_TYPE_CODE.

Item pricing information is also derived in the UNIT_PRICE and CURRENCY_UNIT_PRICE columns. If no autosourcing rules are found for the item, vendor sourcing fails and the UNIT_PRICE is defaulted from the item master for vendor requisition lines and from the CST_ITEM_COSTS_FOR_GL_VIEW for internal requisitions.

The following columns are control columns that the Requisition Import program derives to provide audit trail and relational integrity throughout the interface process:

- CREATION_DATE
- CREATED_BY
- LAST_UPDATE_DATE
- LAST_UPDATED_BY
- LAST_UPDATE_LOGIN
- PROGRAM_ID
- PROGRAM_APPLICATION_ID
- PROGRAM_UPDATE_DATE
- REQUEST_ID

Optional Data

You can use the INTERFACE_SOURCE_CODE and INTERFACE_SOURCE_LINE_ID columns to identify the source of your imported requisitions.

You can enter header, line, and distribution level descriptive flexfield information in the interface table. You can enter up to ten notes for each requisition that you import. The Open Requisitions Interface also lets you enter foreign currency information, project accounting information, UN number, and hazard class information. You can enter the justification for the requisition and indicate whether the requisition is urgent. You can also provide item revision information, source, and destination subinventory information. If you are using requisition encumbrance, you can also provide a USSGL transaction code.

Standard Validation

Your Oracle Purchasing application validates all required columns in the interface table. For specific information on the data implied by these columns, see your *Oracle Purchasing Applications Technical Reference Manual* for details.



Oracle Purchasing Applications Technical Reference Manual

Other Validation

Your Oracle Purchasing application also performs the following cross validations. If a row in the interface table fails validation for any reason the program sets the PROCESS_FLAG in the interface table to 'ERROR' and enters details about every error on that row into the PO_INTERFACE_ERRORS table.

If you enter a SOURCE_TYPE_CODE of 'INVENTORY', the ITEM_ID is required and the item must be stock-enabled for the source organization and internal-order-enabled for the purchasing and destination organizations. The DELIVER_TO_LOCATION_ID must be valid for the destination organization and a customer must be associated with that location in your Oracle Purchasing application. If you also enter a SOURCE_SUBINVENTORY, the item must either be valid in the subinventory, or it must not be restricted to a subinventory. For MRP sourced internal requisitions, the SOURCE_SUBINVENTORY must be a non-nettable subinventory for intra-organization transfers.

If you enter a SOURCE_TYPE_CODE of 'VENDOR' and provide an ITEM_ID, the item must be purchasing-enabled for the purchasing and destination organizations.

If you enter a DESTINATION_TYPE_CODE of 'INVENTORY', the ITEM_ID is required and it must be stock-enabled for the destination organization. If you also enter a DESTINATION_SUBINVENTORY, the item must either be valid in the subinventory or it must not be restricted to a subinventory.

If you enter a DESTINATION_TYPE_CODE of 'SHOP FLOOR', the ITEM_ID is required, and it must be an outside-operation item and purchasing-enabled for the purchasing and destination organizations. The LINE_TYPE_ID must be an outside-operation line type as well.

If you provide a CURRENCY_CODE, the RATE, RATE_DATE, and RATE_TYPE must be provided.

If you are using requisition encumbrance, the GL_DATE that you enter must be in an open or future General Ledger period and an open Purchasing period. Furthermore, if you are using Oracle Inventory, the GL_DATE must be in an open Inventory period for inventory sourced requisitions.

Resolving Failed Requisitions Interface Rows

Error Messages

Your Oracle Purchasing application may display specific error messages during interface processing. Please see the *Oracle Applications Messages Manual* for more details on these messages.



Oracle Applications Messages Manual

Viewing Failed Transactions

You can report on all rows that failed validation by using the Requisition Import Exceptions report. For every transaction in the interface table that fails validation, the Requisition Import Exceptions report lists all the columns that failed validation along with the reason for the failure.

You can identify failed transactions in the Requisitions interface table by selecting rows with a PROCESS_FLAG of 'ERROR'. For any previously processed set of rows identified by INTERFACE_SOURCE_CODE and BATCH_ID, only rows that failed validation remain in the interface table, as all the successfully imported rows are deleted.

For each row in the PO_REQUISITIONS_INTERFACE table that fails validation, the Requisition Import program creates one or more rows with error information in the PO_INTERFACE_ERRORS table.

Fixing Failed Transactions Options

You can update or delete failed rows from the Requisitions interface table using SQL*PLUS.

After you update a row to correct invalid data, you must run the Requisition Import program again to import the row.

You can also delete a failed row and then insert a new, updated row as a replacement.

Rescheduling Requisitions

If you use Oracle Master Scheduling/MRP or a non-Oracle MRP system with Oracle Purchasing, you may find that you need to reschedule requisitions as your planning requirements change. Oracle Purchasing's Requisition Import feature lets you reschedule requisition lines according to changes in your planned orders.

Reschedule Interface Table

You can reschedule requisitions from your planning application with the Reschedule Interface table. Since you have already loaded your requisitions into Oracle Purchasing, you simply need to identify for Oracle Purchasing the requisition lines you want to reschedule. After you identify each line to reschedule, you can update the quantity and the need-by date for the corresponding requisition line. You decide when to import the information from the Requisition Interface table into Oracle Purchasing. Oracle Purchasing lets you use the Reschedule Interface table as often as you want.

Understanding the PO_RESCHEDULE_INTERFACE Table

PO_RESCHEDULE_INTERFACE is the table Oracle Purchasing uses to import information for requisition lines your planning system has rescheduled. One row in the table corresponds to a requisition line whose quantity or need-by date you want to change. Requisition Import updates your requisition lines within Oracle Purchasing with the information in this table. The table PO_RESCHEDULE_INTERFACE consists of columns Oracle Purchasing uses to identify requisition lines for update. The table PO_RESCHEDULE_INTERFACE contains the following columns:

Column Name	Null?	Type
LINE_ID	NOT NULL	NUMBER(15)
QUANTITY		NUMBER

Column Name	Null?	Type
NEED_BY_DATE		DATE
PROCESS_ID		NUMBER(15)
LAST_UPDATE_DATE		DATE
LAST_UPDATED_BY		NUMBER(15)
LAST_UPDATE_LOGIN		NUMBER(15)
CREATION_DATE		DATE
CREATED_BY		NUMBER(15)

Table 9 – 2 (Page 2 of 2)

The following columns are foreign keys to the table and column listed:

Foreign Key	Table	Column
LINE_ID	PO_REQUISITION_LINES	LINE_ID
QUANTITY	PO_REQUISITION_LINES	QUANTITY
NEED_BY_DATE	PO_REQUISITION_LINES	NEED_BY_DATE

Table 9 – 3 (Page 1 of 1)

The column LINE_ID identifies a requisition line which your planning system reschedules. The columns QUANTITY and NEED_BY_DATE contain new information for the requisition lines your planning system updates.

The other columns in the table store the same information the table PO_REQUISITIONS_INTERFACE uses to track when you place data in the table PO_RESCHEDULE_INTERFACE.

Columns Reserved for ReqImport

ReqImport inserts values into the column PROCESS_ID. Requisition Import inserts the PROCESS_ID to identify all requisition lines which you reschedule at one time. You should not insert any data in this column.

Integrating Purge Information with Oracle Payables and Oracle Purchasing Applications

Your Oracle Purchasing and Oracle Payables applications let you purge document and related records from your database. You can easily purge all purchase orders, requisitions, invoices, receipts, and vendors which match your purge criteria, but only if you can purge all corresponding documents. When you purge information from the database, your Oracle Purchasing or Oracle Payables application deletes inactive documents and vendors. Your purchasing or payables application also deletes all references in other tables to documents and vendors that you purge. Your purchasing and payables applications store basic information about the vendors and documents you purge in separate tables.



Purge

(*Oracle Payables Reference Manual* or *Oracle Government Payables Reference Manual*)

Purge

(*Oracle Purchasing Reference Manual* or *Oracle Government Purchasing Reference Manual*)



Submit Purge

(*Oracle Payables Reference Manual* or *Oracle Government Payables Reference Manual*)

Submit Purge

(*Oracle Purchasing Reference Manual* or *Oracle Government Purchasing Reference Manual*)

Below is a list of the tables that contain the summary information of the vendors and documents you successfully purged. You can also review the tables the Purge routines affect when you purge a vendor or a particular type of document. For complete details on all the tables that your Oracle Payables or Oracle Purchasing application uses for purging, please consult the technical reference manuals.



Table Definitions

(*Oracle Payables Applications Technical Reference Manual*)

Table Definitions

(*Oracle Purchasing Applications Technical Reference Manual*)

- FINANCIALS_PURGES – Stores the criteria and status of a single purge process. The table also records the number of records you successfully purge from other tables.
- PO_HISTORY_VENDORS – Stores summary information of the vendors your Oracle Purchasing application purges.
- PO_HISTORY_POS – Stores summary information of the purchase orders your Oracle Purchasing application purges.
- PO_HISTORY_RECEIPTS – Stores summary information of the receipts your Oracle Purchasing application purges.
- PO_HISTORY_REQUISITIONS – Stores summary information of the requisitions your Oracle Purchasing application purges.
- PO_PURGE_PO_LIST – Stores po_header_id's temporarily during the purchase order purge process
- PO_PURGE_REQ_LIST – Stores requisition_header_id's temporarily during the requisition purge process
- PO_PURGE_VENDOR_LIST – Stores vendor_id's temporarily during the vendor purge process
- AP_PURGE_INVOICE_LIST – Stores invoice_id's temporarily during the invoice purge process
- AP_HISTORY_CHECKS – Stores summary information of the invoice payments Oracle Payables purges.
- AP_HISTORY_INVOICES – Stores summary information of the invoices Oracle Payables purges.
- AP_HISTORY_INVOICE_PAYMENTS – Stores the relationship between invoices and payments that Oracle Payables purges.

Tables Affected By Purging a Purchase Order

- PO_ACCRUAL_RECONCILE_TEMP
- PO_ACCEPTANCES
- PO_BLANKET_ITEMS
- PO_DELIVERIES
- PO_DISTRIBUTIONS
- PO_HEADERS
- PO_HEADERS_ARCHIVE
- PO_ITEM_HISTORY
- PO_LINE_LOCATIONS

- PO_LINE_LOCATIONS_ARCHIVE
- PO_LINES
- PO_LINES_ARCHIVE
- PO_NOTE_REFERENCES
- PO_NOTES
- PO_NOTIFICATIONS
- PO_QUALITY_INSPECTIONS
- PO_RECEIPT_HEADERS
- PO_RECEIPTS
- PO_RELEASES

Tables Affected By Purging a Requisition

- PO_APPROVALS
- PO_NOTE_REFERENCES
- PO_NOTES
- PO_REQ_DISTRIBUTIONS
- PO_REQEXPRESS_LINES
- PO_REQUISITION_HEADERS
- PO_REQUISITION_LINES
- PO_NOTIFICATIONS

Tables Affected By Purging an Invoice

- AP_BATCHES
- AP_CHECKS
- AP_DOC_SEQUENCE_AUDIT
- AP_HOLDS
- AP_INVOICES
- AP_INVOICE_DISTRIBUTIONS
- AP_INVOICE_PAYMENTS
- AP_PAYMENT_DISTRIBUTIONS
- AP_PAYMENT_SCHEDULES
- AP_TRIAL_BALANCES

Tables Affected By Purging a Vendor

- PO_VENDORS
- PO_VENDOR_CONTACTS
- PO_VENDOR_SITES
- PO_VENDOR_LIST_ENTRIES
- PO_VENDOR_LIST_HEADERS

Oracle Receivables Applications Open Interfaces

This chapter contains information about the following Oracle Receivables Applications open interfaces:

- Importing Invoice Information into Oracle Receivables Applications Using AutoInvoice
- Processing Bank Receipts Using Oracle Receivables Applications AutoLockbox
- Integrating Customer Information Using Oracle Receivables Applications Customer Interface
- Integrating Oracle Receivables Applications Tax Information Using Sales Tax Rate Interface
- Implementing the Oracle Receivables Applications Tax Vendor Extension

Importing Invoice Information into Oracle Receivables Applications Using AutoInvoice

AutoInvoice is a powerful, flexible tool that lets you import and validate transaction data from other financial systems, from which you can create invoices, debit memos, credit memos and on-account credits. Oracle Receivables and Oracle Government Receivables give you the flexibility to run AutoInvoice together with Customer Interface or separately to meet your business needs.

AutoInvoice is a powerful, flexible tool that lets you import and validate transaction data from other financial systems, from which you can create invoices, debit memos, credit memos and on-account credits. Your Oracle Receivables application gives you the flexibility to run AutoInvoice together with Customer Interface or separately to meet your government needs.

Basic Needs

Oracle Receivables and Oracle Government Receivables provide you with the features you need to satisfy your basic integration needs. You should be able to:

- Automatically convert the data you import into invoices, debit memos, credit memos and on-account credits.
- Choose how you want to uniquely identify the transactions you import.
- Import transaction data from a variety of environments, including your own sales order systems.
- Import transaction data from a variety of environments, including your own order systems.
- Submit individual or groups of transactions for processing using different selection criteria.
- Indicate how you want to order your lines.
- Choose how you want to determine your invoice and accounting dates. You should be able to pass dates from your original system or use a standard algorithm if you do not enter a date.
- Assign default general ledger accounts to your transactions.
- Choose whether you want to pass multiple revenue accounts for each transaction line.

- Assign sales tax or Value Added Tax rates to your transactions. Fully or partially exempt a customer or item from a specific tax code or assign special tax rates to specific items based on the shipping location.
- Assign multiple tax codes and tax accounts to your transactions.
- Review the results of your transaction data import in a clear and concise report. Identify all successfully imported data as well as data that failed the import process. You should be able to review all error messages for each transaction that fails validation.
- Review all messages in the AutoInvoice log file. You can also specify the amount of detail that you want AutoInvoice to display.

Major Features

Simple Integration

AutoInvoice lets you import transaction data from a variety of environments. You can import data from your existing systems, or from any system from which you need data.

Historical Data Conversion

AutoInvoice lets you import transaction data from your previous revenue accounting system, making the transition from your old application to your Oracle Receivables application as smooth and as efficient as possible.

Transaction Flexfields

AutoInvoice provides you with a way to uniquely identify each transaction you import into your Oracle Receivables application. Use Transaction Flexfields to capture information that will help you trace transactions from your Oracle Receivables application back to the systems from which they originated. AutoInvoice ensures that each Transaction Flexfield is unique so you can refer to previously processed transactions. For example, if you are importing a credit memo, you would use the Transaction Flexfield of the credit memo to refer to the transaction being credited. You also use Transaction Flexfields to link transaction lines to other transaction lines and to tax and freight lines.

Creating Transactions

AutoInvoice creates invoices, debit memos, credit memos and on-account credits using the grouping and invoice line ordering rules you specify and answering that your data is valid before creating these transactions.

Calculating Tax

AutoInvoice provides you with the functionality you need to meet your sales tax and other taxing requirements, such as Value Added Tax. You can either pass tax code lines, tax exempt lines or have AutoInvoice automatically determine your tax rates using the hierarchy determined by the tax calculation flow charts. If AutoInvoice determines your tax rates it will take into account any customer or item tax exemptions or item tax exceptions.



Calculating Tax Flow Chart
(*Oracle Receivables Reference Manual* or *Oracle Government Receivables Reference Manual*)

Deriving Invoice and Accounting Dates

AutoInvoice lets you choose how you want to determine invoice and accounting dates for your transactions. Your feeder program can either load these dates directly into the interface tables or, if you leave the date fields empty, your Oracle Receivables application will determine your invoice and accounting dates using a straightforward algorithm.

Passing Multiple Revenue Accounts

Use AutoInvoice to pass multiple revenue accounts and amounts for each of your transactions.

Invoices Against Commitments

AutoInvoice lets you create invoices against commitments in the same way as you would with a manually entered invoice.

Transactions in Closed Accounting Periods

Use AutoInvoice to pass transactions in closed accounting periods. Your Oracle Receivables application automatically uses the first day of the next open accounting period as your default date to determine your accounting distributions.

Complete Validation

AutoInvoice ensures that the data you are importing is accurate and valid within your Oracle Receivables application.

AutoInvoice Processing Report

Your Oracle Receivables application automatically generates the AutoInvoice Processing Report every time you run AutoInvoice, so that you can review the status of your imported transaction data. Your Oracle Receivables application displays summary information about your transactions, such as the number of transaction, sales credit and distribution lines that have passed and failed validation. You can see the number of transactions that have been created in your Oracle Receivables application by currency and transaction class.

Your Oracle Receivables application automatically generates the AutoInvoice Processing Report every time you run AutoInvoice, so you can review the status of your imported transaction data. Your Oracle Receivables application displays summary information about your transactions, such as the number of transaction, revenue credit and distribution lines that have passed and failed validation. You can easily see the number of transactions that have been created in your Oracle Receivables application by currency and transaction class.



AutoInvoice Processing Report
(*Oracle Receivables Reference Manual* or *Oracle Government Receivables Reference Manual*)

Territory Flexfield

Your Oracle Receivables application imports the Territory Flexfield associated with each invoice.

AutoInvoice Validation Report

Your Oracle Receivables application automatically generates the AutoInvoice Validation Report every time you run AutoInvoice, so you can review the error messages for each record that fails validation. You can use this report to help resolve your errors and then re-import your information.



AutoInvoice Validation Report
(*Oracle Receivables Reference Manual* or *Oracle Government Receivables Reference Manual*)

AutoInvoice Purge Program

You have the option to choose if you want to delete data from the AutoInvoice Interface tables once it has been validated and transferred into your Oracle Receivables application. If you want AutoInvoice to automatically delete the data, enter Yes in the Purge Interface Tables field in the Define System Options form. If you want to delete data from the AutoInvoice Interface tables later, enter No in this field. You can choose to run the AutoInvoice Purge program at any time from the Run AutoInvoice form. The AutoInvoice Purge program only deletes data that has been validated and transferred into your Oracle Receivables application.

Definitions

Accounting Flexfield

An innovative, powerful and flexible tool that allows you to define your chart of accounts according to the needs of your organization. You can choose the number, order, and length of each segment.

Accounting Flexfield Segment

One of up to 30 different sections of your Accounting Flexfield, your general ledger account code. Each segment of your code references a value set from the AutoInvoice Interface tables that typically captures one element of your business's structure, such as Company, Fund, Division, Region, or Project.

Accounting Flexfield Segment Value

A series of characters and a description that provides a unique value for a particular value set, such as 0100 for the region value set, meaning eastern Region, or 510 for the division value set, meaning finance Division.

AutoInvoice Interface Tables

A series of three tables from which your feeder system inserts transaction data you transfer from other systems. The fourth table, RA_INTERFACE_ERRORS_ALL is only used by AutoInvoice to store information about interface data that failed validation.

Original System

The external system from which you are transferring data into your Oracle Receivables application tables.

Feeder Program

A program you write to transfer data from an original system into your Oracle Receivables application. The type of feeder program you write depends on the environment from which you are importing data.

Credit Memos

Credits that you give to your customers that refer to a specific transaction. You can create credit memos in the Enter Credit Memos form or through AutoInvoice.

On-Account Credits

Credits that you give to your customers that do not refer to a specific transaction. You can create on-account credits in the Enter Credit Memos form or through AutoInvoice.

Accounting Rules

Rules that AutoInvoice uses to specify revenue recognition schedules for your transactions. You can define an accounting rule where revenue is recognized over a fixed or variable period of time. For example, you can define a fixed duration accounting rule with monthly revenue recognition for a period of 12 months. Accounting Rules are applicable only if the accounting method is 'Accrual'. AutoInvoice will reject all transactions lines that have accounting rules, if the accounting method used is 'Cash Basis'.

Invoicing Rules

Rules that AutoInvoice uses to determine when you bill your invoices. You can bill In Advance or In Arrears. Invoicing Rules are applicable only if the accounting method is 'Accrual'. AutoInvoice will reject all transactions lines that have invoicing rules, if the accounting method used is 'Cash Basis'.

Accounting Rule Start Date

The date your Oracle Receivables application uses for the first accounting entry it creates when you use an accounting rule to recognize revenue. If you choose a variable accounting rule you need to specify a rule duration to let your Oracle Receivables application know how many accounting periods to use this accounting rule.

Invoice Date

The date your Oracle Receivables application prints on an invoice. Your Oracle Receivables application also uses this date to determine the payment due date based on the payment terms you specify on the invoice.

General Ledger Date

Your Oracle Receivables application uses this date to determine the correct accounting period for your transactions. Your Oracle Receivables application's posting program uses this date when interfacing revenue accounting transactions to your general ledger.

Ship Date

The date an item is shipped to your customers. Your feeder program can insert the ship date into the interface tables for items that are able to be shipped. If an item is not able to be shipped (e.g. service), your feeder program should leave the ship date column blank.

Order Date

The date upon which an order for goods or services is entered. Your feeder program can insert the order date into the interface tables.

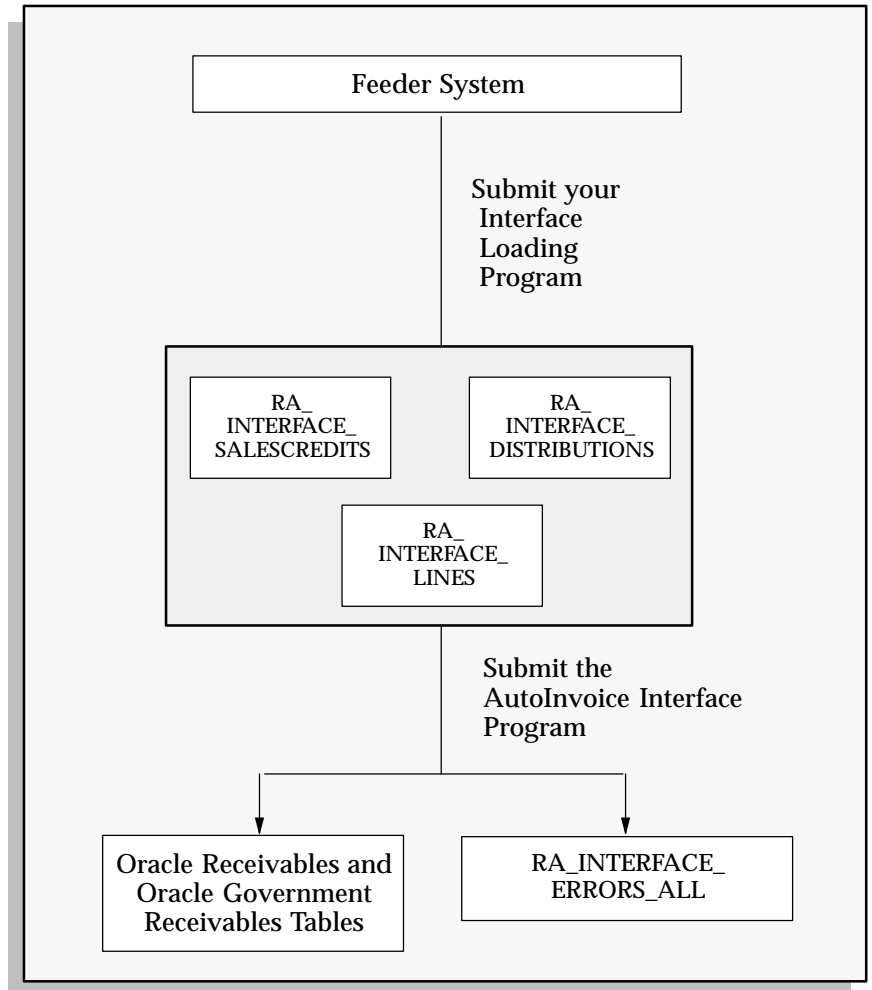
Default Date

A date you specify when invoking AutoInvoice that your Oracle Receivables application uses as the Accounting Rule Start Date, the Invoice Date, and the General Ledger Date, if it cannot determine these dates any other way.

Overview

The following diagram shows how invoice information is imported into your Oracle Receivables application tables.

Figure 10 - 1 Importing invoice information using AutoInvoice



Preparing Oracle Receivables and Oracle Government Receivables

To ensure that AutoInvoice works for you, you should prepare your Oracle Receivables application for any new data that you want AutoInvoice to import.

Accounting Flex Tuning Segment

If you want to increase the performance of AutoInvoice and indices already exist for the GL_CODE_COMBINATIONS table, use the value that you specified for your index as your Accounting Flexfield tuning segment. If you defined a concatenated index use the first column of your concatenated index.

If no indices exist for the GL_CODE_COMBINATIONS table, enter the segment with the most distinct values for your Accounting Flexfield tuning segment. Use the Define System Options form to define your Accounting Flexfield tuning segment.



Define System Options

(Oracle Receivables Reference Manual or Oracle Government Receivables Reference Manual)

System Items Tuning Segment

If you want to increase the performance of AutoInvoice and indices already exist for the MTL_SYSTEM_ITEMS table, use the value that you specified for your index as your System Items Flexfield tuning segment. If you defined a concatenated index use the first column of your concatenated index.

If no indices exist for the MTL_SYSTEM_ITEMS table, enter the segment with the most distinct values for your System Items Flexfield tuning segment. Use the Define System Options form to define your System Items Flexfield tuning segment.



Define System Options

(Oracle Receivables Reference Manual or Oracle Government Receivables Reference Manual)

Territory Tuning Segment

If you want to increase the performance of AutoInvoice and indices already exist for the RA_TERRITORIES table, use the value that you specified for your index as your Territory Flexfield tuning segment. If you defined a concatenated index use the first column of your concatenated index.

If no indices exist for the RA_TERRITORIES table, enter the segment with the most distinct values for your Territory Flexfield tuning

segment. Use the Define System Options form to define your Territory Flexfield tuning segment.



Define System Options
(*Oracle Receivables Reference Manual* or *Oracle Government Receivables Reference Manual*)

SQL Trace

In the Define System Options form specify whether you want to activate SQL trace for AutoInvoice. You might want to use SQL trace for troubleshooting if AutoInvoice is running slow.



Define System Option
(*Oracle Receivables Reference Manual* or *Oracle Government Receivables Reference Manual*)



SQL Trace
(*Oracle 7 Server Administrator's Guide*)

Purge Interface Tables

In the Define System Options form specify whether you want to run the AutoInvoice Purge program immediately after AutoInvoice has completed. If you want to run the AutoInvoice Purge program later, you can access the AutoInvoice Purge program in the Run AutoInvoice form. Your Oracle Receivables application only deletes the records that have successfully transferred into permanent Oracle Receivables and Oracle Government Receivables tables.



Define System Options
Run AutoInvoice
(*Oracle Receivables Reference Manual* or *Oracle Government Receivables Reference Manual*)

Max Memory (in bytes)

In the Define System Options form you can enter the maximum amount of memory that you want to allocate AutoInvoice for validation. Your Oracle Receivables application defaults to 65535 bytes. Enter a lower number if the message, 'Failed to allocate memory for scratch_memory' displays when you run AutoInvoice. Enter a

higher number if the message, 'The given piece of memory is not large enough to hold a single row' displays.



Define System Options

(*Oracle Receivables Reference Manual* or *Oracle Government Receivables Reference Manual*)

Log File Message Level

In the Define System Options form enter a number from 0 to 3, that represents the amount of detail you want displayed in the AutoInvoice log file.

Message Level 0 gives the following entries in the log file:

- Product Version
- Program Name
- AutoInvoice Start Time
- AutoInvoice Concurrent Request Arguments
- Error and Warning Messages
- AutoInvoice End Time
- AutoInvoice Logical Steps

Message Level 1 gives you all of the above entries plus:

- Time-Stamped function labels

Message Level 2 gives you all of the above entries plus:

- Sizes of Allocated Arrays
- Dynamic SQL Statements
- Number of Rows Updated, Inserted and Deleted

Message Level 3 gives you all of the above entries plus:

- Method IV SQL Array Values



Define System Options

(*Oracle Receivables Reference Manual* or
Oracle Government Receivables Reference Manual)



Arrays, Dynamic SQL Statements

(*Programmer's Guide to the ORACLE Precompilers*)

Accounting Flexfield Segment Values

Add Accounting Flexfield segment values to your Oracle Receivables application if your original system uses values not yet defined in your Oracle Receivables application. Enter the name of the Accounting

Flexfield segment for which you want to add a value, and the segment value itself. Be sure to enable the segment value.



Define Key Segment Values
(*Oracle Applications Flexfields Manual*)

Transaction Flexfield

Your Oracle Receivables application uses the Transaction Flexfield to uniquely identify each transaction and transaction line you import through AutoInvoice. Transaction Flexfields are also used to refer to and link transaction lines.

To define the line-level Transaction Flexfield, query 'Line Transaction Flexfield' in the Title field of the Define Descriptive Flexfield Segments form and enter the context and segments associated with this Transaction Flexfield. To define the Transaction Flexfield at the header-level, query 'Invoice Transaction Flexfield' and enter the context and segments associated with this Transaction Flexfield. All segments in the line level transaction flexfield that refer to header information must also exist in the header level transaction flexfield. For example if you define a line-level Transaction Flexfield with 4 segments and only the last 2 segments refer to line-level information, define the header Transaction Flexfield using the first two segments. You must define both the line-level and header-level Transaction Flexfield.

If you do not create a Reference and Link-to transaction flexfields, then your Oracle Receivables application will use your Line Transaction Flexfield structure to link and reference different lines. You do not have to define a separate Reference and Link-to transactions in this case.

However, if you are planning to create a customized form to enter interface data which will display the Reference and Link-to Transaction Flexfields, then you must define Transaction Flexfields in the Define Descriptive Flexfield Segments form. These flexfields must have the same flexfield structures as the line-level Transaction Flexfield.



Define Value Set
Define Descriptive Flexfield Segments
(*Oracle Applications Flexfields Manual*)

Territory Flexfield

If you use territories, you should create your territory flexfield structure before using AutoInvoice.



Define Key Segment Values
(*Oracle Applications Flexfields Manual*)

Line Ordering Rules

Define ordering rules used by AutoInvoice to determine how to order your transaction lines. AutoInvoice randomly orders lines on your transaction if you do not define line ordering rules.



Define Line Order
(*Oracle Receivables Reference Manual* or *Oracle Government Receivables Reference Manual*)

Grouping Rules

Define additional grouping rules or update the default grouping rule provided by your Oracle Receivables application. Grouping rules are used by AutoInvoice to determine how to create your transactions. If you do not define grouping rules in your customer profiles, your Oracle Receivables application uses the grouping rule you defined for your system option. Grouping rules are required if you use AutoInvoice.

If you want to be able to use the information that you pass in your header Transaction Flexfield, then you must group by the segments that make up your header Transaction Flexfield.



Define Grouping Rules
(*Oracle Receivables Reference Manual* or *Oracle Government Receivables Reference Manual*)

Sales Tax Location Flexfield Structure

Define your Sales Tax Location Flexfield structure if you are going to charge your customers location based tax and you do not want to use one of the seeded Location Flexfield structures provided by your Oracle Receivables application.



Define Key Flexfield Segments
(*Oracle Applications Flexfields Manual*)

Locations and Tax Rates

Add or update locations and their associated tax rates if your tax method is set to Sales Tax and your original system uses locations not defined in your Oracle Receivables application. You can either use the Define Tax Locations and Rates form to manually add new locations or the Sales Tax Interface program to import locations and tax rates provided by a tax service.

If your tax method is Value Added Tax and you are validating your customers' addresses, add or update locations if your original system uses locations not defined in your Oracle Receivables application.



Define Tax Locations and Codes

(Oracle Receivables Reference Manual or Oracle Government Receivables Reference Manual)

AutoAccounting

You must set up your Oracle Receivables application's AutoAccounting feature before you run AutoInvoice. AutoAccounting determines default revenue, receivable, freight, tax, unbilled, unearned, and suspense accounts for your invoices.



Define AutoAccounting

(Oracle Receivables Reference Manual or Oracle Government Receivables Reference Manual)

Salesperson/Agent

Add salespersons/agents to your Oracle Receivables application if your original system uses salespersons/agents that are not yet defined in your Oracle Receivables application.



Define Salesperson/Agent

(Oracle Receivables Reference Manual or Government Receivables Reference Manual)

Setup Data

If your original system uses any additional setup data which is not yet defined in your Oracle Receivables application, you need to define this within your Oracle Receivables application before using AutoInvoice. Pay particular attention to the following setup data:

- Add currencies to your Oracle Receivables application if your original system uses currencies not yet defined in your Oracle Receivables application.



Define Period Rates

(Oracle Receivables Reference Manual or Oracle Government Receivables Reference Manual)

- Add or update tax rates assigned to tax codes that are not defined in your Oracle Receivables application.



Define Tax Codes and Rates

(Oracle Receivables Reference Manual or Oracle Government Receivables Reference Manual)

- Add or update tax rates associated with products shipped to specific addresses.



Define Item Tax Rate Exceptions

(Oracle Receivables Reference Manual or Oracle Government Receivables Reference Manual)

- Add or update full or partial customer and item tax exemptions.



Define Product/Customer Exemption

(Oracle Receivables Reference Manual or Oracle Government Receivables Reference Manual)

- Add Freight on Board (FOB) codes to your Oracle Receivables application if your original system uses FOB point codes not yet defined in your Oracle Receivables application. Define FOB point codes in the Define Receivables QuickCodes form with a QuickCodes type of FOB.



Define Receivables QuickCodes

(Oracle Receivables Reference Manual or Oracle Government Receivables Reference Manual)

- Add freight carrier codes to your Oracle Receivables application if your original system uses freight carriers not yet defined in your Oracle Receivables application.



Define Freight Carrier

(Oracle Receivables Reference Manual or Oracle Government Receivables Reference Manual)

- Add payment terms to Oracle Receivables and Oracle Government Receivables if your original system uses payment

terms that are not yet defined in your Oracle Receivables application.



Define Payment Terms

(Oracle Receivables Reference Manual or Oracle Government Receivables Reference Manual)

- Add transaction types to your Oracle Receivables application if your original system uses transaction types not yet defined in your Oracle Receivables application.



Define Transaction Types

(Oracle Receivables Reference Manual or Oracle Government Receivables Reference Manual)

- Add batch sources to your Oracle Receivables application if your original system uses batch sources not yet defined in your Oracle Receivables application.



Define Invoice Sources

(Oracle Receivables Reference Manual or Oracle Government Receivables Reference Manual)

- Add salesperson/agents to your Oracle Receivables application if your original system uses salespersons/agents that are not yet defined in your Oracle Receivables application.



Define Salespersons/Agents

(Oracle Receivables Reference Manual or Oracle Government Receivables Reference Manual)

- Add accounting rules to your Oracle Receivables application if your original system uses accounting rules that are not yet defined in your Oracle Receivables application.



Define Invoicing and Accounting Rules

(Oracle Receivables Reference Manual or Oracle Government Receivables Reference Manual)

- Add units of measure to your Oracle Receivables application if your original system uses units of measure that are not yet defined in your Oracle Receivables application.



Define Units of Measure

(Oracle Receivables Reference Manual or Oracle Government Receivables Reference Manual)

Importing Data From Your Feeder System

Your on-site MIS personnel or Oracle consultant must first write a custom feeder program that transfers transaction data from your original system into your Oracle Receivables application AutoInvoice Interface tables. Your feeder program must convert data from your original system into a standard data format that AutoInvoice can read. AutoInvoice can then convert your imported data into your Oracle Receivables application invoices, credit memos, on-account credits and debit memos.

Writing a Feeder Program

The type of environment from which you want to transfer your data determines the type of feeder program you need to write. For example, you can use SQL*Loader, SQL*Report, PL/SQL, or Pro*C to write a feeder program to transfer transaction data from a non-Oracle system. Or, you can write a conversion program to transfer historical data from your previous accounting system.

Selecting an Import Utility

SQL*Loader and SQL*Report are powerful and easy-to-use tools that should be able to accommodate all of your import needs. However, depending on the complexity of your import program, you may also want to use Oracle's Pro* language products such as Pro*C, Pro*Cobol and Pro*Fortran to write the program.

Understanding the Interface Tables

Oracle Receivables and Oracle Government Receivables uses three tables to temporarily store data you transfer from other systems. These tables are called RA_INTERFACE_LINES_ALL, RA_INTERFACE_SALESCREDITS_ALL and RA_INTERFACE_DISTRIBUTIONS_ALL. The fourth table, RA_INTERFACE_ERRORS_ALL is used only by AutoInvoice to store information about interface data that failed validation. For a detailed description of each column in these tables, please refer to the Table and Column Description section.

AutoInvoice Validation

AutoInvoice validates your data for compatibility with your Oracle Receivables application. It ensures that the columns in your Oracle Receivables application's Interface tables reference the appropriate values and columns in your Oracle Receivables application. To see the validation AutoInvoice performs for each column in the AutoInvoice tables, please refer to the column description section in this essay.

Existence

For some columns, AutoInvoice ensures that the values are already defined in your Oracle Receivables application. However, AutoInvoice does not validate against any effectivity date or status.

Batch Sources

You do not have to pass values for all of the fields that are referenced in the Define Invoice Sources form. If you want AutoInvoice to ignore any of these values for a specific batch source you can set the field to 'None' in the form.

Uniqueness

AutoInvoice ensures that the invoice number you supply is unique within a given batch source and the document number you supply is unique within the associated sequence type.

AutoInvoice also ensures that the Transaction Flexfield you supply is unique.

Precision

AutoInvoice ensures that the amount and the accounted amount you supply have the correct precision for a given currency.

Cross Validation

AutoInvoice ensures that certain column values agree with each other. These values can be within an interface table or multiple interface tables.

For example, if you specify in your batch source that you do not want to use accounting rules, AutoInvoice ignores any values you supply for invoicing rule, accounting rule and accounting rule duration.

Using AutoInvoice

Running AutoInvoice

You submit AutoInvoice using the Run AutoInvoice form. If AutoInvoice converts your transaction data into the required data format, and all of the data passes validation in your Oracle Receivables application, then you can run AutoInvoice in one step. However, if your feeder program loads the interface tables with invalid data, AutoInvoice informs you of the validation errors in both the AutoInvoice Processing and AutoInvoice Validation reports. In this case, you need to correct any errors by modifying data in the interface tables and then rerun AutoInvoice on the corrected data.

Oracle Receivables and Oracle Government Receivables let you use transactions created by AutoInvoice as you would those entered manually through Oracle Receivables and Oracle Government Receivables form.

Reviewing the AutoInvoice Execution and Validation Reports

Use the AutoInvoice Execution Report to review summary information about your transactions. AutoInvoice automatically creates this report every time you run AutoInvoice. Each report lists the total number of transaction, sales credit and distribution lines that were successfully imported, as well as those that failed.

Use the AutoInvoice Execution Report to review summary information about your transactions. AutoInvoice automatically creates this report every time you run AutoInvoice. Each report lists the total number of transaction, revenue credit and distribution lines that were successfully imported, as well as those that failed.

Note: It is possible to have the number of Successfully Processed lines less than the number Selected and have no lines that Failed Validation. This will occur when a Credit Memo for an Invoice and the Invoice are submitted in the same batch, and the Credit Memo is selected first. Since the Invoice has not been processed yet, the Credit Memo will go unprocessed during this import but not fail. The unprocessed Credit Memo remains in the interface table and will be processed the next time you submit AutoInvoice. The Interface Lines section of the execution report will appear as follows:

Selected: 9
Successfully Processed: 8
Failed Validation: 0

AutoInvoice also automatically generates the AutoInvoice Validation Report if you have records that failed validation. This report displays all error messages associated with each transaction, sales credit and distribution line that failed validation. You can use this information to identify which transactions need fixing.



AutoInvoice Execution Report
AutoInvoice Validation Report
(*Oracle Receivables Reference Manual* or *Oracle Government Receivables Reference Manual*)



Oracle Applications Messages Manual

Correcting Errors

You can correct errors based on the error messages you receive in the AutoInvoice Validation Report. You may need to make changes in your Oracle Receivables application or in your feeder program. For example, if you receive an error message explaining that the salesperson you specified for an invoice does not exist in your Oracle Receivables application, you can add the salesperson to your Oracle Receivables application or fix your feeder program to only transfer salespersons which your Oracle Receivables application recognizes.

You can correct errors based on the error messages you receive in the AutoInvoice Validation Report. You may need to make changes in your Oracle Receivables application or in your feeder program. For example, if you receive an error message explaining that the agent you specified for an invoice does not exist in your Oracle Receivables application, you can add the agent to your Oracle Receivables application or fix your feeder program to only transfer agents which your Oracle Receivables application recognizes.

Passing Payment Methods and Customer Bank Accounts

All references to parent customer information in this section are only applicable if the Bill To customer only has one parent and the

relationship is not reciprocal. For example, if the Bill To customer for the line has more than one parent, 1 & 2 will not apply.

Payment Methods

Regardless if you are passing manual or automatic payment methods, AutoInvoice validates that the payment method belongs to the Bill-To customer/site or the parent of the Bill-To customer/site, if it has one. Additionally, the payment method must have at least one bank account in the currency of the transaction or its' Receipts Multi-Currency flag set to Yes.

If you do not pass a payment method, AutoInvoice defaults one using the following hierarchy:

1. Primary payment method assigned to the primary site for the parent
2. Primary payment method assigned to the parent customer
3. Primary payment method assigned to the Bill To site for the line
4. Primary payment method assigned to the Bill To customer for the line

Customer Bank Accounts

If you are passing a customer bank account and the payment method associated with the transaction is automatic, AutoInvoice validates that the customer bank account belongs to one of the following, otherwise the line is rejected:

1. Bank account assigned to the primary site for the parent
2. Bank account assigned to the parent customer
3. Bank account assigned to the Bill To site for the line
4. Bank account assigned to the Bill To customer for the line

If you do not pass a customer bank account and the payment method associated with the transaction is automatic, AutoInvoice defaults one using the following hierarchy:

1. Primary bank account assigned to the primary site for the parent
2. Primary bank account assigned to the parent customer
3. Primary bank account assigned to the Bill To site for the line
4. Primary bank account assigned to the Bill To customer for the line

If AutoInvoice is unable to default a customer bank account, the line is

rejected.

AutoInvoice uses the customer bank account to determine whether the paying customer is the parent or the Bill To customer. If the paying customer is the Bill To customer, the paying site is the Bill To site. If the paying customer is the parent, the paying site is the primary Bill To site of the parent. Customer bank accounts are not used for manual payment methods.

Freight Lines

AutoInvoice lets you pass freight lines as individual transactions or as references to other transactions. The `RA_INTERFACE_LINES_ALL.LINK_TO_LINE_ATTRIBUTE1-15` and `RA_INTERFACE_LINES_ALL.LINK_TO_LINE_CONTEXT` columns determine whether a freight line will become an individual freight only transaction, or part of another transaction.

If you want to pass a freight line that refers to another transaction line, enter the Line Transaction Flexfield of the transaction you want this freight line to refer. To pass freight lines `RA_INTERFACE_LINES_ALL.LINE_TYPE` must be set to 'FREIGHT'.

If you want to pass a freight-only line, enter a Line Transaction Flexfield that refers to a 'dummy' line. This 'dummy' line must have a value in `RA_INTERFACE_LINES_ALL.MEMO_LINE_ID` or `RA_INTERFACE_LINES_ALL.MEMO_LINE_NAME` and the memo line must have `AR_MEMO_LINES_ALL.LINE_TYPE = 'FREIGHT'`. In addition the Quantity, Unit Price, and Amount fields for this line must be null or zero.

Using AutoAccounting for Freight

If AutoAccounting for Freight is based on Standard Lines, you will not be able to import invoices with header level freight. All freight lines in this case must be associated to a standard line for AutoAccounting to determine the account. If the transaction has a line type of "LINE" with an inventory item of freight, "FRT", AutoAccounting will use the accounting rules for the freight type account rather than the revenue type account.

Importing Multiple Header Freight Lines

AutoInvoice ensures that there is at most one freight line for an imported invoice, or at most one freight line per transaction line, but

not both. If multiple header freight lines applied to one invoice have been imported, AutoInvoice will validate that all of the freight lines apply to the same freight account and consolidate them to one line. This consolidated freight line will be the only freight line for this invoice that is passed to the core receivables tables. If all of the freight lines do not apply to the same freight account the invoice will be rejected.

Audit Trail for Consolidated Freight Lines

The log file generated by AutoInvoice will list the freight attributes for auditing purposes:

- customer_trx_id
- interface_line_id of the freight line chosen for consolidation
- sum of the freight amounts

Tax

AutoInvoice gives you flexibility to handle all of your taxing needs. If your tax method is VAT you can either pass tax lines through the AutoInvoice interface tables or have your Oracle Receivables application automatically calculate your tax lines for you. If you are using Sales Tax, your Oracle Receivables application will always calculate tax for you. However you can choose to pass additional tax lines with tax codes of type VAT or Sales Tax.

Passing Tax Lines Through AutoInvoice

AutoInvoice lets you pass tax lines as individual transactions or as references to other transactions. If you are passing tax lines you can only pass tax lines associated to tax codes of type VAT or Sales Tax. The RA_INTERFACE_LINES_ALL. LINK_TO_LINE_ATTRIBUTE1-15 and RA_INTERFACE_LINES_ALL. LINK_TO_LINE_CONTEXT columns will determine whether a tax line will become an individual tax only transaction, or part of another transaction.

If you want to pass a tax line that refers to another transaction line, enter the Line Transaction Flexfield of the transaction you want this tax line to refer. To pass tax lines

RA_INTERFACE_LINES_ALL.LINE_TYPE must be set to 'TAX'.

If you want to pass a tax-only line, enter a Line Transaction Flexfield that refers to a 'dummy' line. This 'dummy' line must have a value in

RA_INTERFACE_LINES_ALL.MEMO_LINE_ID or RA_INTERFACE_LINES_ALL.MEMO_LINE_NAME and the memo line must have AR_MEMO_LINES.LINE_TYPE = 'TAX'. In addition the Quantity, Unit Price, and Amount fields for this line must be null or zero.

Tax lines with precedence numbers can be passed through AutoInvoice by providing a value for the TAX_PRECEDENCE column. For example, to associate 5 tax lines with an invoice line where one line is non-precedent, 2 lines have a precedence of 1, and the remaining 2 are precedence 2. Your interface table values for the line type, tax code, and tax precedence columns would look like this:

LINE_TYPE	TAX_CODE	TAX_PRECEDENCE
TAX	CODE1	null
TAX	CODE2	1
TAX	CODE3	1
TAX	CODE4	2
TAX	CODE5	2

Table 10 – 1 (Page 1 of 1)

Calculating Tax

Certain criteria must be fulfilled before AutoInvoice will calculate tax. The following flowcharts show the hierarchy your Oracle Receivables application uses to calculate tax. Use the flowchart titled 'Tax Method = Sales Tax' if you entered 'Sales Tax' in the Tax Method field of the Define System Option forms. Use the flowchart titled 'Tax Method = Value Added Tax' if you entered 'Value Added Tax'. If a Ship To address does not exist for a transaction, your Oracle Receivables application derive tax from the Bill To information. For example, if a transaction has a Ship To customer, but no Ship To address, your Oracle Receivables application use the Bill To information along with the appropriate hierarchy to determine the tax.

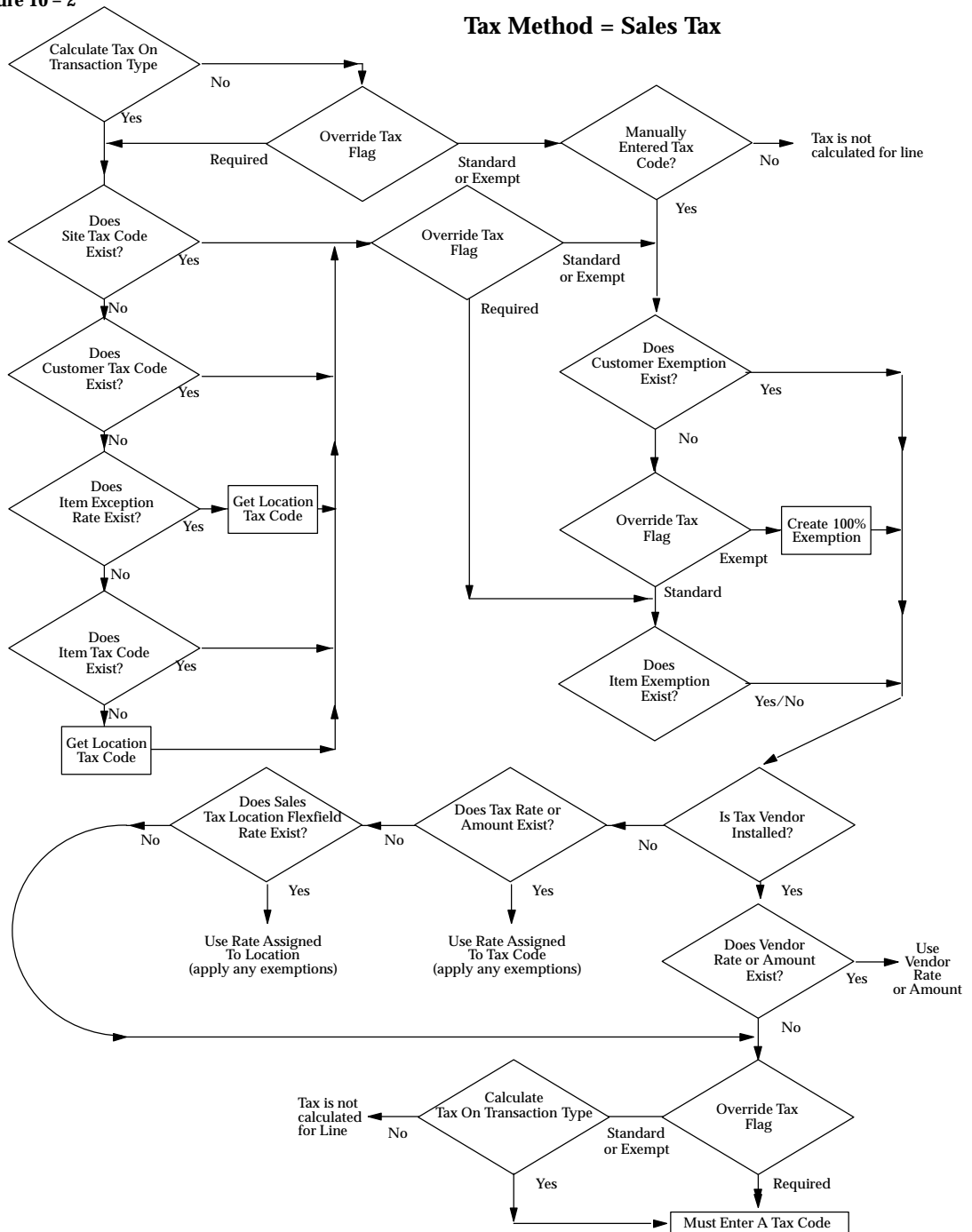
Legend for Sales Tax Hierarchy Diagram System Option		
	Value	Action
Use Tax code for Customer Site	No	Skip Level
Use Tax code for Customer	No	Skip Level
Use Tax code for Product	No	Skip Level

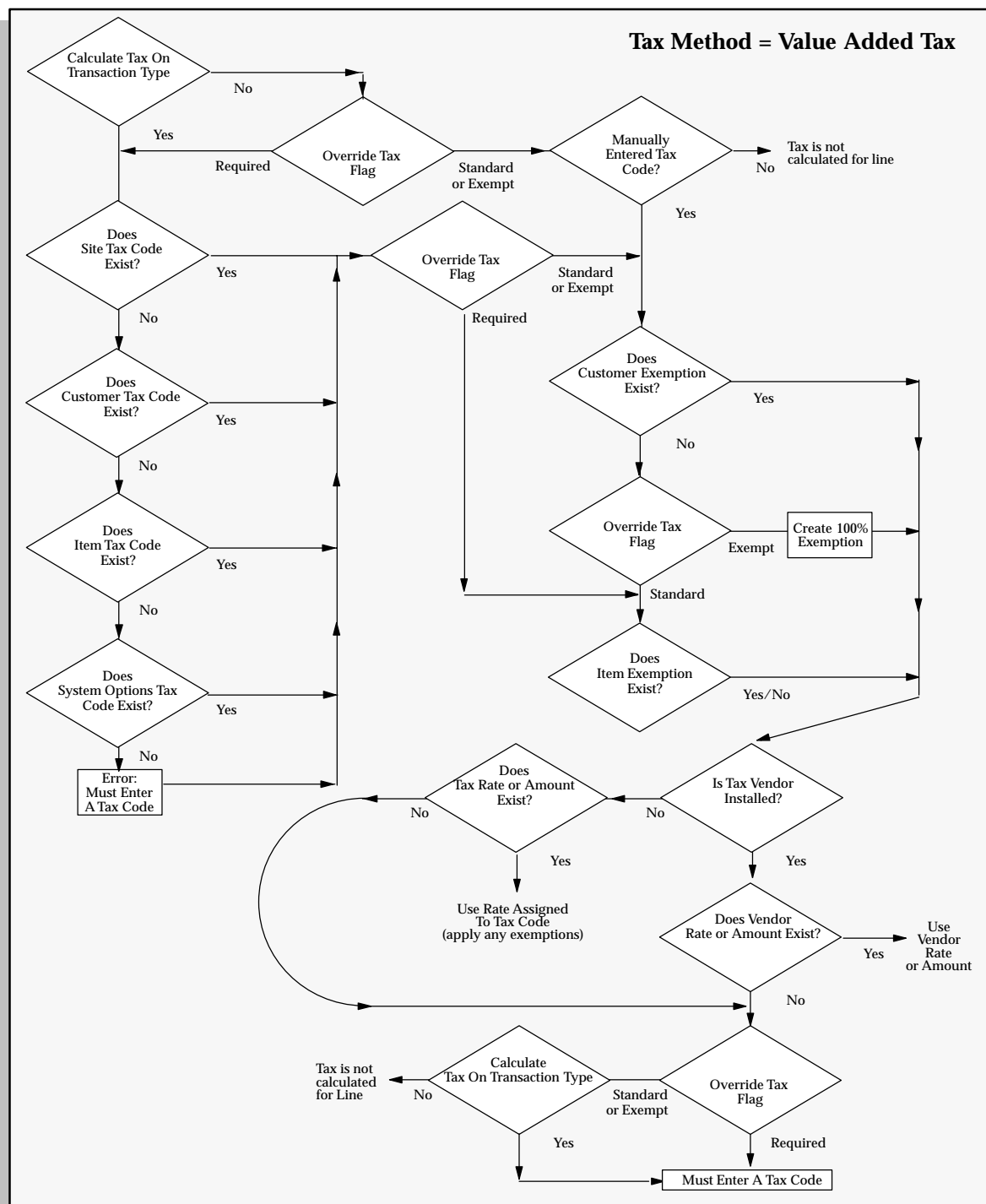
Legend for Sales Tax Hierarchy Diagram		
System Option	Value	Action
Use Item for Tax Rate Exception	No	Skip Level
Use Customer Exemptions	No	Skip Level
Use Product Exemptions	No	Skip Level

Table 10 – 2 (Page 2 of 2)

Figure 10 - 2

Tax Method = Sales Tax





Legend for Value Added Tax Hierarchy Diagram		
System Option	Value	Action
Use Tax code for Customer Site	No	Skip Level
Use Tax code for Customer	No	Skip Level
Use Tax code for Product	No	Skip Level
Use Customer Exemptions	No	Skip Level
Use Product Exemptions	No	Skip Level

Table 10 – 3 (Page 1 of 1)

Use the table below to see what tax information needs to be passed to the interface tables to achieve the desired results.

Desired Result	Line Type	Tax Code	Tax Rate/Tax Amount	Tax Exempt Flag	Tax Exempt Number	Tax Exempt Reason Code or Meaning	Comments
Oracle Receivables and Oracle Government Receivables should calculate the tax based on the standard tax logic	Line – No Tax line associated with this line	NULL	NULL	NULL or 'S'	NULL	NULL or 'S'	If you have not passed any tax lines with the invoice lines, and the tax exempt flag is NULL or 'S', Oracle Receivables and Oracle Government Receivables will calculate tax for you.
You want Oracle Receivables and Oracle Government Receivables to calculate Sales tax, but want to pass additional tax code	Tax	Of type VAT or Sales Tax and must be adhoc	Must pass either the tax rate or amount	NULL or 'S'	NULL	NULL	The invoice line will have 2 tax lines. The first one will be location based tax calculated by Oracle Receivables and Oracle Government Receivables. Second will be the tax line passed through AutoInvoice.

Table 10 – 4 (Page 1 of 2)

Desired Result	Line Type	Tax Code	Tax Rate/Tax Amount	Tax Exempt Flag	Tax Exempt Number	Tax Exempt Reason Code or Meaning	Comments
You want to exempt the invoice line from any taxes and your system option 'Use Customer Exemptions' is set to Yes	Line	NULL	NULL	'E'	Pass tax exemption number	Pass reason for exemption	If the tax exemption number does not exist on file, Oracle Receivables and Oracle Government Receivables will create an unapproved exemption on the fly. There will be no tax calculated on this invoice line.
You want to enforce tax on an invoice line, even if any exemptions exist on the file	Line	NULL	NULL	'R'	NULL	NULL	Oracle Receivables and Oracle Government Receivables calculates tax as per its standard logic, ignoring any exemptions.

Table 10 – 4 (Page 2 of 2)

Sales Tax

Sales tax is calculated by AutoInvoice using the tax rates associated with your shipping address. It will only be calculated for shipping addresses which are in the country defined in the Country field of the Define System Options form. Your Oracle Receivables application lets you pass exception rates and exemptions for customers or items. Sales Tax lines cannot be passed into AutoInvoice tables.



Calculating Tax

(Oracle Receivables Reference Manual or Oracle Government Receivables Reference Manual)

Other Tax Codes

If you do not want AutoInvoice to calculate tax based on location you can pass tax codes through lines with line_type = 'Tax'. Tax codes can be of type VAT or Sales Tax only and must be adhoc. You must also pass either a tax rate or amount with the code. Any exemptions must be calculated into the rate or amount. For more information on tax

codes and tax exemptions please refer to the Calculating Tax Topical Essay.



Calculating Tax
(*Oracle Receivables Reference Manual* or *Oracle Government
Receivables Reference Manual*)

Finance Charges

AutoInvoice processes debit memos with finance charges lines and credit memos that are against debit memos with finance charges lines.

If `LINE_TYPE = 'CHARGES'`, AutoInvoice does not calculate tax, freight or sales credits on this line. Also, if you are passing your finance charges distribution in `RA_INTERFACE_DISTRIBUTIONS_ALL`, `ACCOUNT_CLASS` must be `= 'CHARGES'`.

In order for AutoInvoice to pass a finance charge line do not enter value for the following columns in `RA_INTERFACE_LINES_ALL`:

`INVOICING_RULE_ID`
`INVOICING_RULE_NAME`
`ACCOUNTING_RULE_ID`
`ACCOUNTING_RULE_NAME`
`ACCOUNTING_RULE_DURATION`
`RULE_START_DATE`
`UOM_CODE`
`UOM_NAME`
`AMOUNT`

If you are passing a debit memo finance charges line `RA_INTERFACE_LINES_ALL.QUANTITY` must = 1. If you are passing a credit memo against a debit memo with a finance charges line `RA_INTERFACE_LINES_ALL.QUANTITY` must = -1 or 1.

Account Assignments

AutoInvoice lets you determine how to assign general ledger accounts to transactions you import through AutoInvoice. You can either pass

your accounts through the AutoInvoice Interface tables or have AutoAccounting determine them. You can even pass some of your accounts and have AutoAccounting determine the rest.

Passing Account Information

If you choose to pass your accounts, AutoInvoice looks at the batch source to determine whether to expect Accounting Flexfield segment values or id's.

If you pass segment values you must assign values to RA_INTERFACE_DISTRIBUTIONS_ALL.SEGMENT1-30. Only assign values to enabled segments. For example, if you enable six Accounting Flexfield segments, you must assign values in SEGMENT1-6.

If you pass IDs you must enter the code combination id of the Accounting Flexfield in RA_INTERFACE_DISTRIBUTIONS_ALL.CODE_COMBINATION_ID.

If you want the option of AutoInvoice dynamically inserting code combinations, you must pass segments.

Using AutoAccounting

If you want AutoAccounting to determine your general ledger accounts you must not enter values in RA_INTERFACE_DISTRIBUTIONS_ALL. AutoInvoice will determine all of your accounts using information you pass for each line. Use the Define AutoAccounting form to define your revenue, receivables, tax, freight, clearing, unbilled receivable, and unearned revenue accounts.

Note: If AutoAccounting for Freight is based on Standard Lines, you will not be able to import invoices with header level freight. If the transaction has a line type of "LINE" with an inventory item of freight, "FRT", AutoAccounting will use the accounting rules for the freight type account rather than the revenue type account.

Note: If AutoAccounting is set up to derive its segments from Salesreps, then you must pass rows in RA_INTERFACE_SALESCREDITS_ALL for each invoice line in RA_INTERFACE_LINES_ALL. This is true even if your system option: Require Salesreps is set to No.

Transaction Flexfields

Transaction flexfields are descriptive flexfields used by AutoInvoice to uniquely identify transaction lines. Because they are unique for each transaction line, they can also be used to reference and link to other lines. Your Oracle Receivables application lets you determine how you want to build your transaction flexfield structure and what information you want to capture.

There are four types of transaction flexfields.

- Line Transaction Flexfield
- Reference Transaction Flexfield
- Link-To Transaction Flexfield
- Invoice Transaction Flexfield

You must define the Line Transaction flexfield. AutoInvoice always uses the Line Transaction flexfield structure for both the Link-to and Reference information when importing invoices. You need to explicitly define the Link-to, Reference and Invoice Transaction flexfield structures only if this information is to be displayed on a custom form.

Your Oracle Receivables application gives you the option of displaying transaction flexfield information for imported invoices, in lists of invoice values throughout the product. Use the System Profile Option AR: Transaction Flexfield QuickPick to select the Transaction Flexfield Segment you want to display. For example, if you want to be able to reference the order number for imported invoices when using any Invoice list of values, you need to assign the transaction flexfield segment that holds the order number to the AR: Transaction Flexfield QuickPick profile option. The order number will now display in the reference column of all Invoice lists for imported invoices.

Line Transaction Flexfield

Use columns INTERFACE_LINE_ATTRIBUTE1-15 and INTERFACE_LINE_CONTEXT to define the line transaction flexfield. Line transaction flexfields are unique for each record in the interface table and therefore can be used as record identifiers.

Reference Transaction Flexfield

Reference Transaction flexfields have the same structure as the Line Transaction flexfields.

These are used to refer to previously processed transactions. For example, you might want to import a credit memo and apply it to an invoice or associate an invoice to a specific commitment. To refer a credit memo to a specific invoice use the REFERENCE_LINE_ATTRIBUTE1-15 and REFERENCE_LINE_CONTEXT columns of the credit memo to enter the Line Transaction flexfield of the invoice.

Link-To Transaction Flexfield

Link-To Transaction flexfields also have the same structure as the Line Transaction flexfield.

Use Link-To Transaction flexfields to link transaction lines together in the interface table. For example, you might want to import tax and freight charges that are associated to specific transaction lines. If you want to associate a specific tax line with a specific transaction line, use the LINK_TO_LINE_ATTRIBUTE1-15 and LINK_TO_LINE_CONTEXT columns of the tax line to enter the Line Transaction flexfield of the invoice.

Invoice Transaction Flexfields

Create a new flexfield with a similar structure as the Line Transaction flexfield, but only include header level segments. For example, if the Line Transaction flexfield structure has four segments and the last two segments contain line level information, define your Invoice Transaction flexfield using the first two segments only. Segments included in the Invoice Transaction Flexfield should be included in the AutoInvoice grouping rules.

Transaction Flexfields: An example

The following example will illustrate how records are linked in the interface table using the Link-To or the Reference Transaction flexfield columns:

Consider an invoice against a commitment. This invoice has 2 line records, 1 header freight record and one tax record. The records will be represented in the interface table as follows:

Line Type	Line Flex			Link-To Flex			Reference Flex			Ref ID
	Context	1	2	Context	1	2	Context	1	2	
Line	Service	A	1							✓
Line	Service	A	2							✓
Freight	OE	A	T1							
Tax	Service	A	3	Service	A	1				

Table 10 – 5 (Page 1 of 1)

Notes:

- The transaction type for records of an invoice is INV.
- Each record in the interface table is uniquely identified by its Line Transaction flexfield. In the above example 2 segments have been enabled for the Line Transaction flexfield service context. Therefore the combination of context plus the 2 segments should be unique.
- Tax records always have to be linked to a line record. In the above example, the tax record is linked to the first line record, Service A 1.
- Since the freight is at the header level, it is not linked to any line record via the Link-To Transaction flexfield.
- Records with different contexts can be grouped together into one invoice.
- If the invoice is against an existing commitment, populate the Reference_line_id (Ref ID) column with the unique identifier (customer_trx_line_id) of the commitment.

Consider a credit memo that credits the freight and the first line of the previous invoice. It will be represented in the interface table as follows:

Line Type	Line Flex			Link-To Flex			Reference Flex			Ref ID
	Context	1	2	Context	1	2	Context	1	2	
Freight	OE	A	T2				OE	A	T1	
Line	OE	A	T3				Service	A	1	

Table 10 – 6 (Page 1 of 1)

Notes:

- The transaction type credit memos is CM.
- Link credit memos to invoices either via the Reference Transaction flexfield or the reference_line_id. The reference_line_id is the unique id within Oracle Receivables.

An on-account credit will be represented as follows:

Line Type		Line Flex Context		Link-To Flex Context			Reference Flex Context			Ref ID
		1	2	1	2		1	2		
Line	Service	B	1							

Table 10 – 7 (Page 1 of 1)

Notes:

- The transaction type for on-account credits is CM.
- AutoInvoice assumes all records with transaction type CM to be on-account credits if there are no values filled in the reference_line_id column (Ref ID) or the Reference Transaction flexfield.

Indexing Transaction Flexfields

We advise you create indexes on your Transaction Flexfield columns if you want to query transaction flexfield information in your invoice headers and lines. Additionally, without the indexes, the validation portions of the AutoInvoice program could be slow. You should define unique, concatenated indexes on the following tables and columns that you use for your Transaction Flexfield header and line information:

Table	Columns
RA_CUSTOMER_TRX_LINES_ALL	interface_line_attribute1-15
RA_CUSTOMER_TRX_ALL	interface_header_attribute1-15
RA_INTERFACE_LINES_ALL	interface_line_attribute1-15

Navigate to the Define Descriptive Flexfield Segments form, and query your Line Transaction Flexfield. Note each context of this Flexfield and, for each context, note which segments are enabled using interface line attribute columns from the RA_INTERFACE_LINES_ALL table.

You should then create unique, concatenated indexes for the same interface line attribute columns in the RA_CUSTOMER_TRX_LINES_ALL and RA_INTERFACE_LINES_ALL tables and for the same interface header attribute columns in the RA_CUSTOMER_TRX_ALL table.

If you just have one context defined then you only need to create one index for each table mentioned above. However, if you have multiple contexts defined you may want to create multiple indexes per table. Use the example below to help you decide how to set up your indexes.

Suppose your Line Transaction flexfield has three contexts which are set up as follows:

Flexfield Context	Attribute Columns assigned to Enabled Segments
Context1	Interface_line_attribute1
Context1	Interface_line_attribute2
Context2	Interface_line_attribute1
Context2	Interface_line_attribute2
Context2	Interface_line_attribute3
Context3	Interface_line_attribute3
Context3	Interface_line_attribute9

Table 10 – 8 (Page 1 of 1)

You should define the combination of indexes which best meets your needs. In the example above, you could create three indexes per table. One for each context. Alternatively, you could create just two indexes. One for context3 and another for context1. The latter would be used by context2 because contexts1 and 2 have the same first two attribute columns.

Use the following syntax for your Create Index Statement:

```

$ sqlplus <AR username>/<AR password>
SQL> CREATE [UNIQUE] INDEX index ON
      {Table (column1, column2, ...)}
      |CLUSTER cluster}
      |INITRANS n] [MAXTRANS n]
      [TABLESPACE tablespace]
      [STORAGE storage]
      [PCTFREE n]
      [NOSORT];

```

Using Grouping Rules to Create Transactions

AutoInvoice uses grouping rules to determine how to create invoices, debit memos and credit memos. Grouping rules contain various transaction attributes that must be identical for the same transaction. For example, if transaction number is part of your grouping rule and you have two records in the interface tables with different transaction numbers, AutoInvoice will create two transactions. Your Oracle Receivables application provides you with two different types of transaction attributes, required and optional. You cannot add or drop required transaction attributes, but you can always add optional ones. Below is a list of mandatory and optional attributes. (From the table RA_INTERFACE_LINES_ALL)

Mandatory Attributes

AGREEMENT_ID
 COMMENTS
 CONVERSION_DATE
 CONVERSION_RATE
 CONVERSION_TYPE
 CREDIT_METHOD_FOR_ACCT_RULE
 CREDIT_METHOD_FOR_INSTALLMENTS
 CURRENCY_CODE
 CUSTOMER_BANK_ACCOUNT_ID
 CUST_TRX_TYPE_ID
 DOCUMENT_NUMBER
 DOCUMENT_NUMBER_SEQUENCE_ID

GL_DATE
HEADER_ATTRIBUTE1-15
HEADER_ATTRIBUTE_CATEGORY
INITIAL_CUSTOMER_TRX_ID
INTERNAL_NOTES
INVOICING_RULE_ID
ORIG_SYSTEM_BILL_ADDRESS_ID
ORIG_SYSTEM_BILL_CONTACT_ID
ORIG_SYSTEM_BILL_CUSTOMER_ID
ORIG_SYSTEM_SHIP_ADDRESS_ID
ORIG_SYSTEM_SHIP_CONTACT_ID
ORIG_SYSTEM_SHIP_CUSTOMER_ID
ORIG_SYSTEM_SOLD_CUSTOMER_ID
ORIG_SYSTEM_BATCH_NAME
PREVIOUS_CUSTOMER_TRX_ID
PRIMARY_SALESREP_ID
PRINTING_OPTION
PURCHASE_ORDER
PURCHASE_ORDER_DATE
PURCHASE_ORDER_REVISION
REASON_CODE
RECEIPT_METHOD_ID
RELATED_CUSTOMER_TRX_ID
SET_OF_BOOKS_ID
TERM_ID
TERRITORY_ID
TRX_DATE
TRX_NUMBER
Optional Attributes
ACCOUNTING_RULE_DURATION

ACCOUNTING_RULE_ID
ATTRIBUTE1-15
ATTRIBUTE_CATEGORY
INTERFACE_LINE_ATTRIBUTE1-15
INTERFACE_LINE_CONTEXT
INVENTORY_ITEM_ID
ORG_ID
REFERENCE_LINE_ID
RULE_START_DATE
SALES_ORDER
SALES_ORDER_DATE
SALES_ORDER_LINE
SALES_ORDER_REVISION
SALES_ORDER_SOURCE
TAX_CODE
TAX_RATE

If you have transactions that fail validation, your Oracle Receivables application looks at the value you entered in the Invalid Lines field for your batch source to determine the grouping of your transactions. If you entered Reject Invoice, AutoInvoice rejects all of the transactions that make up one invoice if any of the transactions are invalid. For example, if your grouping rule specifies that three transactions should be created as one invoice and one of the transactions has an error, AutoInvoice rejects all three transactions and does not create an invoice. However, if you entered Create Invoice, AutoInvoice rejects the one invalid transaction and creates an invoice from the two remaining valid transactions.

Transaction Number Validation

Your Oracle Receivables application validates that transaction and document numbers are unique within a batch after grouping has completed. In certain cases AutoInvoice will create multiple invoices in the same group with the same transaction or document number. Once grouping is completed AutoInvoice checks for duplicate transaction and document numbers and reports any lines that fail validation. For example, two lines are imported with the same transaction number but

have different currency codes. These lines will be split into two separate invoices during grouping due to the different currency codes. Once grouping has completed, both of the invoices will fail validation due to identical transaction numbers.

Using Line Ordering Rules

Line ordering rules are used by AutoInvoice to determine how to order and number each line after your transactions have been grouped into invoices, debit memos and credit memos. You can specify a line ordering rule for each grouping rule. You might want to use line ordering rules to ensure that the highest invoice line amounts are listed first. In this case, define a line ordering rule where amount is your transaction attribute and descending is your order by type. Oracle Receivables and Oracle Government Receivables provides you with the following transaction attributes that you can use in your line ordering rules: (From the table RA_INTERFACE_LINES_ALL)

ACCOUNTING_RULE_DURATION

ACCOUNTING_RULE_ID

ACCOUNTING_RULE_NAME

AMOUNT

ATTRIBUTE_CATEGORY

ATTRIBUTE1-15

FOB_POINT

INTERFACE_LINE_ATTRIBUTE1-15

INTERFACE_LINE_CONTEXT

RA_INTERFACE_ERRORS_ALL

QUANTITY

QUANTITY_ORDERED

REASON_CODE

REASON_CODE_MEANING

REFERENCE_LINE_ATTRIBUTE1-15

REFERENCE_LINE_CONTEXT

REFERENCE_LINE_ID

SALES_ORDER
SALES_ORDER_DATE
SALES_ORDER_LINE
SALES_ORDER_SOURCE
SHIP_DATE_ACTUAL
SHIP_VIA
TAX_CODE
UNIT_SELLING_PRICE
UNIT_STANDARD_PRICE
UOM_CODE
UOM_NAME
WAYBILL_NUMBER

Importing Invoices With Rules

Use AutoInvoice to import invoices with accounting and invoicing rules if your accounting method is 'Accrual'. AutoInvoice rejects all invoices with rules if the accounting method is 'Cash Basis'. Accounting rules determine the accounting period(s) in which the revenue distributions for an invoice line are recorded. Invoicing rules determine the accounting period in which the receivable amount is recorded. Your Oracle Receivables application provides you with two invoicing rules, 'Bill in Advance' and 'Bill in Arrears'. You supply AutoInvoice with the model account which contains the accounting distributions and the percent allocated to each account. You must run the Revenue Recognition Program before your Oracle Receivables application can create your accounting entries. See the example below for the effects of using accounting and invoicing rules through AutoInvoice. Assume that you have already run the Revenue Recognition Program for each accounting period.

Example #1

Invoice #101

Transaction Amount: **\$300**

(RA_INTERFACE_LINES_ALL.QUANTITY (3)*

RA_INTERFACE_LINES_ALL.UNIT_SELLING_PRICE (\$100))

Accounting Rule: **Monthly**
(RA_INTERFACE_LINES_ALL.ACCOUNTING_RULE_ID)

Invoicing Rule: **Bill in Advance**
(RA_INTERFACE_LINES_ALL.INVOICING_RULE_ID)

Duration (Number of Periods): **3**
(RA_INTERFACE_LINES_ALL.ACCOUNTING_RULE_DURATION)

Rule Start Date: **1/1/XX**
(RA_INTERFACE_LINES_ALL.RULE_START_DATE)

Payment Term: **Net 30**
(RA_INTERFACE_LINES_ALL.TERM_ID)

Oracle Receivables and Oracle Government Receivables creates the following accounting entries:

1/1/XX	DR Accounts Receivable	300	
	CR Unearned Revenue		200
	CR Revenue		100
2/1/XX	DR Unearned Revenue	100	
	CR Revenue		100
3/1/XX	DR Unearned Revenue	100	
	CR Revenue		100

In the above example, the transaction date for this invoice is 1/1/XX, with a payment due date of 1/31/XX. If we had chosen an invoicing rule of 'Bill in Arrears', the transaction date in the above example would have been 3/1/XX with a payment due date of 3/31/XX.

Importing Credit Memos

When you are importing credit memos against transactions, AutoInvoice ensures that the Open Receivables flag of the credit memo being imported must match the Open Receivables flag of the transaction it is crediting.

Credit Memos Against Invoices With Rules

When you import credit memos against invoices with rules, AutoInvoice uses the method you entered in `RA_INTERFACE_LINES_ALL.CREDIT_METHOD_FOR_ACCT_RULE` to determine how to reverse the accounting entries created for the original invoice. You can either enter 'LIFO', 'PRORATE', or 'UNIT'. If you choose 'LIFO', AutoInvoice reverses the accounting entries beginning with the last period. If you choose 'PRORATE', AutoInvoice prorates the credit amount across all accounting periods. If you choose 'UNIT', AutoInvoice lets you credit specific quantities, starting with the period specified in the column `RA_INTERFACE_LINES_ALL.LAST_PERIOD_TO_CREDIT` and working backwards.



Understanding Credit Memos
(*Oracle Receivables and Oracle Government Receivables Reference Manual or Oracle Government Receivables Reference Manual*)

Credit Memos Against Invoices Without Rules

When you import credit memos against invoices without rules, AutoInvoice first uses the general ledger date in the interface table as the general ledger date of the credit memo. If you do not pass a general ledger date, AutoInvoice uses the default date you specified in the Run AutoInvoice form. The credit memo lines must always have the same general ledger date as the credit memo.

The credit memo general ledger date must be equal to or greater than the general ledger date of the invoice you are crediting. Also, the credit memo general ledger date must be in an 'Open' or 'Future' period.

Credit memos against invoices without rules that are imported through AutoInvoice behave the same as those entered manually through the Enter Credit Memos form. For example, you pass the amount you want to credit and your Oracle Receivables application automatically creates all the accounting reversal entries. Your Oracle Receivables application also automatically reverses the sales revenue and non-revenue credit you assigned to your salespeople.

Credit memos against invoices without rules that are imported through AutoInvoice behave the same as those entered manually through the Enter Credit Memos form. For example, you pass the amount you want to credit and your Oracle Receivables application automatically creates all the accounting reversal entries. Your Oracle Receivables

application also automatically reverses the revenue and non-revenue credit you assigned to your agents.

Credit Memos Against Tax and Freight Lines

When you import credit memos, AutoInvoice ensures that you do not overapply your tax and freight lines.

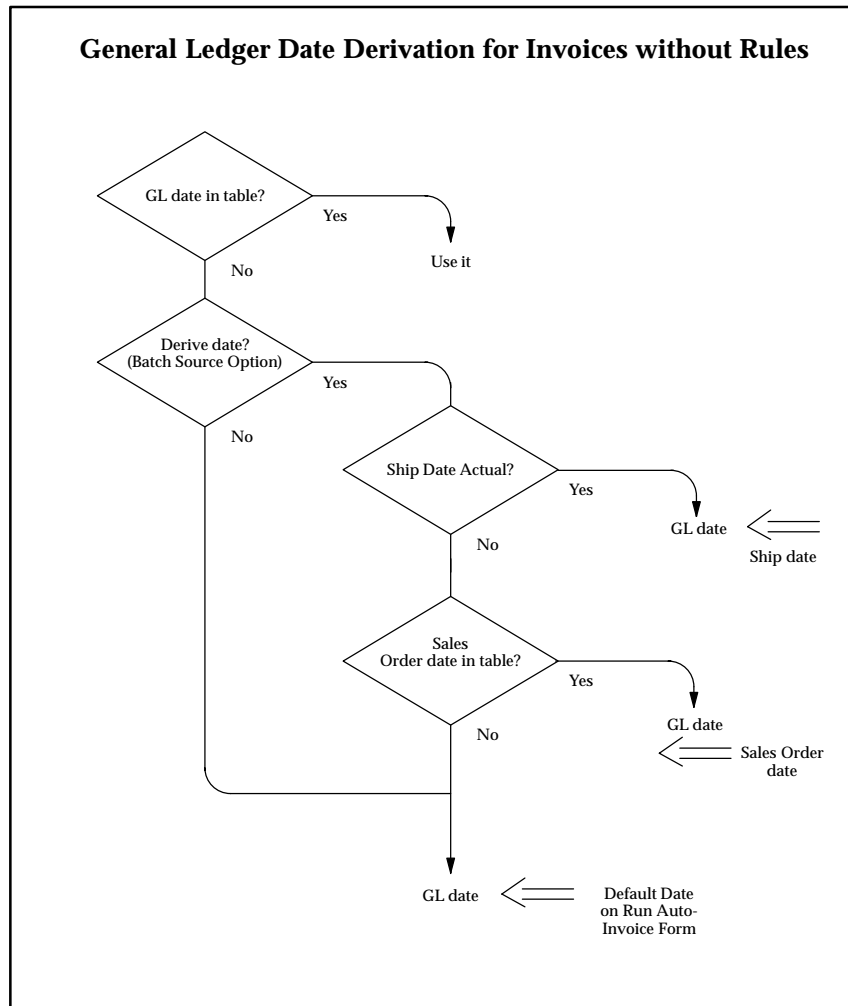
Determining Dates

Determining General Ledger Dates for Invoices Without Rules

If your invoice does not use rules, AutoInvoice first uses the general ledger date in the interface table, if one exists. If you did not pass a general ledger date and you elected not to derive the date, AutoInvoice uses the date you entered in the Run AutoInvoice form.

If you elect to derive the general ledger date, AutoInvoice first uses the ship date in the interface table. If the ship date does not exist, AutoInvoice uses the sales order date. If the sales order date does not exist, AutoInvoice uses the date you entered in the Run AutoInvoice form.

Figure 10 – 3 General Ledger Date Derivation for Invoices without rules



Determining General Ledger Dates for Invoices With Rules

If your invoice uses Bill in Advance as the invoicing rule, AutoInvoice uses the earliest accounting rule start date as the general ledger date of the invoice.

If your invoice uses Bill in Arrears as the invoicing rule and the invoice line has an accounting rule of type 'Accounting, Fixed Duration' and a period of 'Specific Date', AutoInvoice computes an ending date using the latest accounting rule date.

For all other accounting rules, AutoInvoice computes an ending date for each invoice line based on the accounting rule, accounting rule start

date and duration. Once AutoInvoice computes the ending date for each line of your transaction it takes the latest date and uses it as the general ledger date of the invoice.

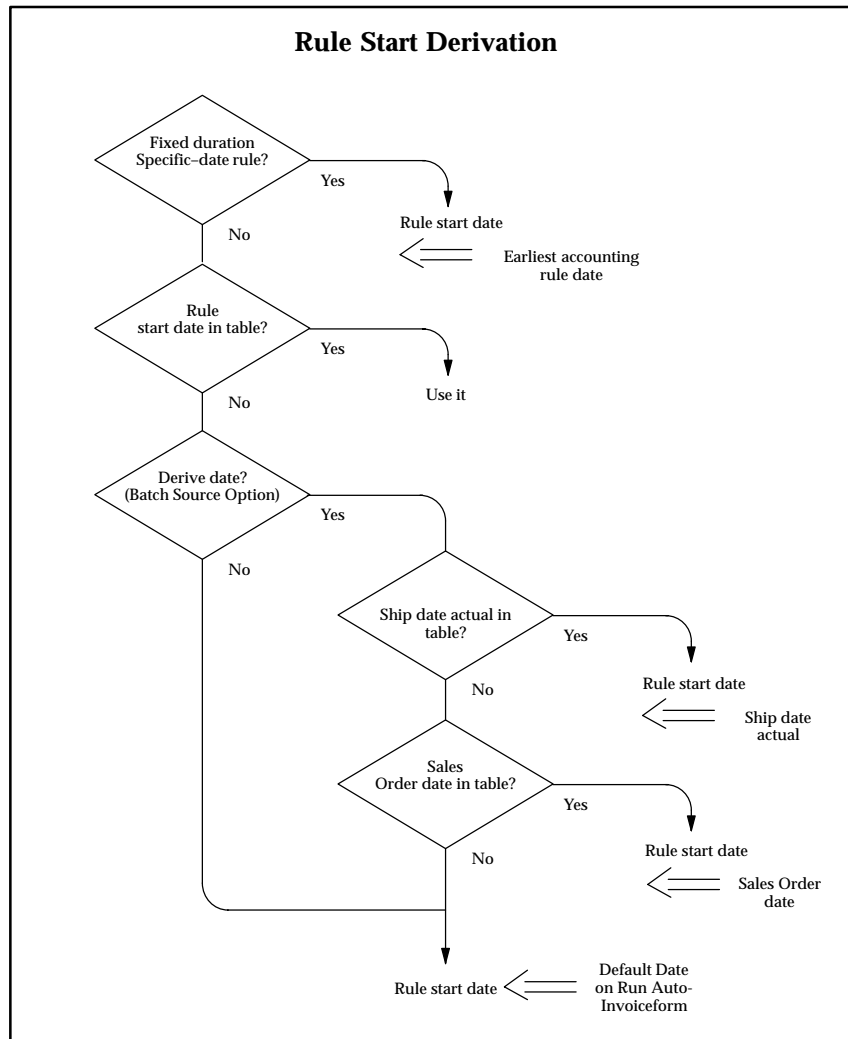
Rule Start Date

Your Oracle Receivables application uses the default date specified in the Run AutoInvoice form if your invoice does not use a fixed duration accounting rule with a specific date period or you have not elected to derive the rule start date.

If your invoice has an accounting rule with a type of 'Accounting, Fixed Duration' and a period of 'Specific Date', AutoInvoice uses the earliest accounting rule date as your rule start date. For example, if your accounting rule dates are 10-JUN-93, 10-JUL-93 and 10-AUG-93, AutoInvoice uses 10-JUN-93 as your rule start date.

If you elected to derive the rule start date, AutoInvoice first uses the ship date in the interface table. If the ship date does not exist, AutoInvoice uses the sales order date. If the sales order date does not exist, AutoInvoice uses the date you entered in the Run AutoInvoice form.

**Figure 10 – 4 Rule
Start Derivation**



Determining Credit Memo Dates

If a transaction date is not passed for your credit memo, AutoInvoice uses the following hierarchy to determine the credit memo date: credit memo general ledger date, and the general ledger date for the invoice's receivable distribution or the date on the Run AutoInvoice form, whichever is later.

If a general ledger date is not passed, AutoInvoice uses the general ledger date for the invoice's receivable distribution or the date on the Run AutoInvoice form, whichever is later.

Determining the Transaction Dates

If a transaction date is not passed for your invoice or debit memo, AutoInvoice uses the general ledger date.



Suggestion: If you use Oracle Inventory and Oracle Order Entry for sales order shipments, you should elect to derive your dates, and use the shipment date for your invoice general ledger date. In this way you can ensure that you have booked your revenue and cost to the same accounting period.

If you do not match revenue and cost in the same period, you violate basic GAAP principles, and may distort your profit. In addition, you are unable to run a meaningful Margin Analysis Report. This report summarizes your revenue and cost of goods sold transactions by item and customer order, and specifies a transaction date range. If your transactions are booked in the wrong period, this report reflects those incorrect transactions.



Margin Analysis Report
(*Oracle Inventory Reference Manual*)

Validating Dates

AutoInvoice uses the following logic when validating general ledger and rule start dates that you either pass or are determined by AutoInvoice. If you use time stamps when you enter dates (i.e. 31-Jul-92 23:59:00), AutoInvoice will remove the time stamp prior to validation.

General Ledger Dates

AutoInvoice rejects lines if the:

- Accounting period for the general ledger date is not defined.
- General ledger date is in a 'Closed' or 'Not Opened' period and the GL Date in a Closed Period field for your batch source is set to 'Reject'. (For invoices that use Bill in Arrears rules, AutoInvoice only rejects lines that have a general ledger date in a 'Closed' period)
- General ledger date of the credit memo is before the invoice general ledger date and/or the credit memo date is before the invoice date.

Rule Start Dates

AutoInvoice rejects lines if the:

- Rule start date for lines that used Bill in Advance rules are in 'Closed' or 'Not Opened' periods and the GL Date in a Closed Period field for your batch source is set to 'Reject' or if the accounting period for the rule start date is not defined.
- Rule start date for lines that used Bill in Arrears rules results in a general ledger date in a 'Closed' period and the GL Date in a Closed Period field for your batch source is set to 'Reject' or if the accounting period for the general ledger date is not defined.
- Rule start date is not the earliest date specified for your accounting rule and you are passing an accounting rule with a type of 'Accounting, Fixed Duration' and a period of 'Specific Date'.

Other Validation for Lines With Rules

Besides validating dates, AutoInvoice also validates and rejects lines if:

- The accounting rule has overlapping periods.
- All of the accounting periods do not exist for the duration of your accounting rule.

Adjusting General Ledger Dates

If the GL Date in a Closed Period field for your batch source is set to 'Adjust' and you pass a general ledger date in a 'Closed' or 'Not Opened' period for a line that does not use rules or uses Bill in Advance rules, AutoInvoice will adjust the general ledger date. AutoInvoice also adjusts the general ledger date if this batch source field is set to 'Adjust' and your pass a line that uses Bill in Arrears rules and the general ledger date falls into a 'Closed' period. AutoInvoice uses the following rules in the order listed when adjusting general ledger dates:

- AutoInvoice first uses the last day of the last prior period, if this period has a status of 'Open'.
- If a prior period with at status of 'Open' does not exist, AutoInvoice uses the first day of the first subsequent period that

was a status of 'Open'. If there are more than one subsequent periods that have a status of 'Open', AutoInvoice cannot adjust the general ledger date and the line is rejected.

- If an 'Open' period does not exist, AutoInvoice uses the first day of the first subsequent period that has a status of 'Future'. If it cannot find a future period, AutoInvoice cannot adjust the general ledger date and the line is rejected.

Oracle Receivables and Oracle Government Receivables Tables

AutoInvoice transfers transaction data into the following Oracle Receivables and Oracle Government Receivables transaction tables:

RA_BATCHES
RA_CUSTOMER_TRX
RA_CUSTOMER_TRX_LINES
RA_CUST_TRX_LINE_GL_DIST
RA_CUST_TRX_LINE_SALESREPS
AR_PAYMENT_SCHEDULES
AR_RECEIVABLE_APPLICATIONS
AR_ADJUSTMENTS

Table and Column Descriptions

Below is a detail description of the three interface tables your Oracle Receivables application uses to store transaction data from your original system. Each column has important detail information you need to know to successfully run AutoInvoice. The fourth table, RA_INTERFACE_ERRORS_ALL is only used by AutoInvoice to store information about interface data that failed validation.

RA_INTERFACE_LINES_ALL

Column Name	Value
ACCOUNTING_RULE_DURATION	Enter the accounting rule duration for this transaction.

If LINE_TYPE = 'LINE' or you are passing header freight, and this transaction uses a variable duration accounting rule, you must enter a value in this column.

If LINE_TYPE = 'TAX' , 'CHARGES' or you are passing freight for a specific line, do not enter a value in this column.

For credit memos and on-account credits, do not enter a value in this column.

Validation: Accounting periods must be defined for the duration of the accounting rule in GL_PERIODS and RA_INTERFACE_LINES_ALL.GL_DATE and RA_INTERFACE_LINES_ALL.RULE_START_DATE must be in a period that has a status of 'Open' or 'Future'
The value in this column must be a positive integer.

Destination: RA_CUSTOMER_TRX_LINES.ACCOUNTING_RULE_DURATION

ACCOUNTING_ RULE_ID

Enter the accounting rule id for this transaction.

If LINE_TYPE = 'LINE' or you are passing header freight, this column is optional. For invoice lines with rules, you must enter either a value in this column or in ACCOUNTING_RULE_NAME, depending on the value you entered for your batch source. If you entered a value in ACCOUNTING_RULE_NAME, AutoInvoice defaults a value in this column.

If LINE_TYPE = 'TAX', 'CHARGES' or you are passing freight for a specific line, do not enter a value in this column.

For credit memos do not enter a value in this column, AutoInvoice uses the accounting rule from the transaction you are crediting.

Validation: Must exist in RA_RULES.RULE_ID and RA_RULES.TYPE = 'A' or 'ACC_DUR'. If LINE_TYPE = 'CHARGES' then this column must be null.

Destination: RA_CUSTOMER_TRX_LINES.ACCOUNTING_RULE_ID

ACCOUNTING_ RULE_NAME

Enter the accounting rule name for this transaction.

If LINE_TYPE = 'LINE' or you are passing header freight, this column is optional. For invoice lines with rules, you must enter either a value in this column or in ACCOUNTING_RULE_ID, depending on the value you entered for your batch source

If LINE_TYPE = 'TAX', 'CHARGES', or you are passing freight for a specific line, do not enter a value in this column.

For credit memos do not enter a value in this column. AutoInvoice uses the accounting rule from the transaction you are crediting.

Validation: Must exist in RA_RULES.NAME and RA_RULES.TYPE = 'A' or 'ACC_DUR'. If LINE_TYPE = 'CHARGES' then this column must be null.

Destination: None

ACCTD_AMOUNT

Do not enter a value. Oracle Receivables and Oracle Government Receivables does not currently use this column.

Validation: None

Destination: None

AGREEMENT_NAME

Enter the name of the customer agreement for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES' or you are passing header freight, this column is optional. Depending on the value you entered for your batch source you can enter either a value in this column or in AGREEMENT_ID. For invoice lines against a commitment, AutoInvoice will default the agreement from the commitment if AGREEMENT_NAME and AGREEMENT_ID are null and a agreement exists for the commitment.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line do not enter a value in this column.

For credit memos do not enter a value in this column, AutoInvoice uses the customer agreement from the transaction you are crediting.

Validation: Must exist in SO_AGREEMENTS.NAME

Destination: None

AGREEMENT_ID

Enter the customer agreement id for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES' or you are passing header freight,

this column is optional. Depending on the value you entered for your batch source you can enter either a value in this column or in AGREEMENT_NAME. For invoice lines against a commitment, AutoInvoice will default the agreement from the commitment if AGREEMENT_NAME and AGREEMENT_ID are null and a agreement exists for the commitment.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line do not enter a value in this column.

For credit memos do not enter a value in this column, AutoInvoice uses the customer agreement from the transaction you are crediting.

Validation: Must exist in SO_AGREEMENTS.ID

Destination: None

AMOUNT

Enter the revenue amount for this transaction.

If LINE_TYPE = 'LINE' and this transaction is not a freight-only nor a tax-only line, you must enter a value in this column. If this transaction is a dummy line for freight-only or tax-only, do not enter a value in this column. AutoInvoice ignores any values you enter in this column if this transaction is a dummy line.

If LINE_TYPE = 'TAX', a value must be entered in either this column or the tax_rate column. Any exemptions must be factored into either of the two columns.

If LINE_TYPE = 'FREIGHT' and you are passing either header freight or freight for a specific line, you must enter a value in this column.

If LINE_TYPE = 'CHARGES', do not enter a value in this column.

For credit memos and on-account credits, enter the credit amount for this transaction.

Validation: If LINE_TYPE = 'CHARGES' then this column must be null. AutoInvoice will correct revenue amounts that have the wrong currency precision

Destination: If you are not using suspense
RA_CUSTOMER_TRX_LINES.REVENUE_
AMOUNT and RA_CUSTOMER_TRX_LINES.
EXTENDED_AMOUNT

If you are using suspense

RA_CUSTOMER_TRX_LINES.
REVENUE_AMOUNT

ATTRIBUTE1-15

Enter the Descriptive Flexfield attribute information for this transaction. Descriptive Flexfield attributes allow you to store additional columns, the contents of which you define. These columns are optional.

Validation: None

Destination: RA_CUSTOMER_TRX_LINES.ATTRIBUTE1-15

ATTRIBUTE_ CATEGORY

Enter the Descriptive Flexfield category information for this transaction. Descriptive Flexfield categories allow you to store different categories of attributes. This column is optional.

Validation: None

Destination: RA_CUSTOMER_TRX_LINES.ATTRIBUTE_
CATEGORY

BATCH_SOURCE_ NAME

Enter the name of the batch source for this transaction. AutoInvoice uses your batch source to determine your transaction and batch numbering method and your AutoInvoice processing options. You must enter a value in this column.

Validation: Must exist in RA_BATCH_SOURCES.NAME and
RA_BATCH_SOURCES.BATCH_SOURCE_TYPE =
'FOREIGN'

Destination: RA_BATCHES.BATCH_SOURCE_ID and
RA_CUSTOMER_TRX.BATCH_SOURCE_ID

COMMENTS

Enter comments about this transaction.

If LINE_TYPE = 'LINE', 'CHARGES' or you are passing header freight, this column is optional.

If LINE_TYPE = 'TAX' or your are passing freight for a specific line, do not enter text in this column.

Validation: None

Destination: RA_CUSTOMER_TRX.COMMENTS

CONVERSION_ DATE

Enter the exchange rate date for this transaction. If you do not enter a date, AutoInvoice defaults to the transaction date. If the currency of

the transaction line is the same as the base currency, then leave this column null.

Validation: None

Destination: RA_CUSTOMER_TRX.EXCHANGE_DATE

CONVERSION_ RATE

Enter the exchange rate for this transaction.

If CONVERSION_TYPE is User, you MUST enter a value in this column. Otherwise do not enter a value. If the currency of the transaction is the same as the base currency, enter 'User' and set CONVERSION_RATE to 1.

Validation: If
RA_INTERFACE_LINES_ALL.CONVERISON_TY
PE = 'User' then this column must not be null,
otherwise it must be null.

Destination: RA_CUSTOMER_TRX.EXCHANGE_RATE

CONVERSION_ TYPE

Enter the exchange rate type for this transaction. If the currency of the transaction is the same as the base currency, enter 'User' and set CONVERSION_RATE to 1. You must enter a value in this column.

Validation: Must exist in GL_DAILY_CONVERSION_TYPES.
CONVERSION_TYPE

Destination: RA_CUSTOMER_TRX.EXCHANGE_RATE_TYPE

CREDIT_METHOD_ FOR_ACCT_RULE

Enter the credit method for crediting a transaction which uses an accounting rule. Choices include PRORATE, LIFO, or UNIT.

If this transaction is a credit memo against a transaction which uses an accounting rule and LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, you must enter a value in this column.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column. AutoInvoice will ignore any value that you enter in this column.

For on-account credits do not enter a value in this column.

Validation: Must be either 'PRORATE', 'LIFO', 'UNIT' or
NULL

Destination: RA_CUSTOMER_TRX.CREDIT_METHOD_FOR_
RULES

CREDIT_ METHOD_FOR_ INSTALLMENTS

Enter the credit method for crediting a transaction which uses split payment terms. Choices include PRORATE, LIFO, or FIFO.

If this transaction is a credit memo against a transaction which uses split payment terms and LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, you may enter a value in this column. If you do not enter a value, AutoInvoice defaults to PRORATE.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column. AutoInvoice will ignore any value that you enter in this column.

For on-account credits do not enter a value in this column.

Validation: Must be either 'PRORATE', 'LIFO', 'FIFO' or NULL

Destination: RA_CUSTOMER_TRX.CREDIT_METHOD_FOR_INSTALLMENTS

CURRENCY_CODE

Enter the currency code for this transaction. You must enter a value in this column.

For credit memos enter the currency code of the invoice you are crediting.

Validation: Must exist in FND_CURRENCIES.CURRENCY_CODE

Destination: RA_CUSTOMER_TRX.INVOICE_CURRENCY_CODE and AR_PAYMENT_SCHEDULES.INVOICE_CURRENCY_CODE

CUSTOMER_BANK_ ACCOUNT_ID

Enter the Bill-To customer bank account id for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional. Depending on the value you entered for your batch source you can enter either a value in this column or in CUSTOMER_BANK_ACCOUNT_NAME. If you entered a value in CUSTOMER_BANK_ACCOUNT_NAME, AutoInvoice defaults a value in this column.

If the payment method is of type 'Automatic', and this column is NULL, AutoInvoice will default a value for you. Please refer to the section on Passing Payment Methods and Customer Bank Accounts for more details on how AutoInvoice defaults and validates customer banks.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

Validation: Must exist in
AP_BANK_ACCOUNTS.BANK_ACCOUNT_ID

If the payment method is of type 'Manual',
RA_INTERFACE_LINES_ALL.CUSTOMER_BANK_
K_
ACCOUNT_ID must be NULL

Destination: RA_CUSTOMER_TRX.CUSTOMER_BANK_
ACCOUNT_ID

CUSTOMER_BANK_ ACCOUNT_NAME

Enter the Bill-To customer bank account name for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional. Depending on the value you entered for your batch source you can enter either a value in this column or in CUSTOMER_BANK_ACCOUNT_ID.

If the payment method is of type 'Automatic', and this column is NULL, AutoInvoice will default a value for you. Please refer to the section on Passing Payment Methods and Customer Bank Accounts for more details on how AutoInvoice defaults and validates customer banks.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

Validation: AP_BANK_ACCOUNTS.BANK_ACCOUNT_
NAME

If the payment method is of type 'Manual',
RA_INTERFACE_LINES_ALL.CUSTOMER_BANK_
K_
ACCOUNT_NAME must be NULL

Destination: None

CUSTOMER_TRX_ID

This column is used by AutoInvoice and should be left null. AutoInvoice defaults a value into this column using your grouping rules.

Validation: None

Destination: RA_CUSTOMER_TRX.CUSTOMER_TRX_ID,
AR_PAYMENT_SCHEDULES.CUSTOMER_TRX_ID,
RA_CUSTOMER_TRX_LINES.CUSTOMER_TRX_ID
and
RA_CUST_TRX_LINE_GL_DIST.CUSTOMER_TRX_ID

CUST_TRX_TYPE_ID Enter the transaction type id for this transaction.

This column is optional but depending on the value you entered for your batch source you must enter either a value in this column or in CUST_TRX_TYPE_NAME. If you entered a value in CUST_TRX_TYPE_NAME, AutoInvoice defaults a value in this column.

For invoice lines against a commitment, AutoInvoice defaults the invoice transaction type from the transaction type of the commitment if CUST_TRX_TYPE_ID and CUST_TRX_TYPE_NAME are null.

For credit memos you must enter the id of the credit memo transaction type which has been assigned to the transaction you are crediting.

Validation: Must exist in
RA_CUST_TRX_TYPES.CUST_TRX_TYPE_ID

Destination: RA_CUSTOMER_TRX.CUST_TRX_TYPE_ID

CUST_TRX_TYPE_NAME Enter the transaction type name for this transaction.

This column is optional but depending on the value you entered for your batch source you must enter either a value in this column or in CUST_TRX_TYPE_ID.

For invoice lines against a commitment, AutoInvoice defaults the invoice transaction type from the transaction type of the commitment if CUST_TRX_TYPE_ID and CUST_TRX_TYPE_NAME are null.

For credit memos you must enter the name of the credit memo transaction type which has been assigned to the transaction you are crediting.

Validation: RA_CUST_TRX_TYPES.NAME

Destination: None

DESCRIPTION	<p>This is a required column in AutoInvoice. Enter the description for this transaction.</p> <p>Validation: None</p> <p>Destination: RA_CUSTOMER_TRX_LINES.DESCRPTION</p>
DOCUMENT_ NUMBER	<p>Enter the document number for this transaction.</p> <p>If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight and the creation method for the sequence numbering of this transaction is Manual, you must enter a value in this column.</p> <p>If LINE_TYPE = 'LINE', 'CHARGES' or you are passing header freight and the creation method is Automatic, do not enter a value in this column, AutoInvoice will create a unique document number.</p> <p>If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.</p> <p>Validation: Number must not already exist in Oracle Receivables and Oracle Government Receivables</p> <p>Destination: RA_CUSTOMER_TRX.DOC_SEQUENCE_VALUE</p>
DOCUMENT_ NUMBER_ SEQUENCE_ID	<p>This column is used by AutoInvoice and should be left null. AutoInvoice uses this column to store the document sequence id for this transaction.</p> <p>Validation: None</p> <p>Destination: RA_CUSTOMER_TRX.DOC_SEQUENCE_ID</p>
EXCEPTION_ID	<p>This column is used by AutoInvoice and should be left null. AutoInvoice defaults a value in this column when a tax exception occurs.</p> <p>If your transaction is a credit memo, AutoInvoice defaults the tax exemption id of the transaction you are crediting.</p> <p>Validation: None</p> <p>Destination: RA_CUSTOMER_TRX_LINES.ITEM_EXCEPTION_RATE_ID</p>
EXEMPTION_ID	<p>This column is used by AutoInvoice and should be left null. AutoInvoice defaults a value in this column when this transaction is partially or fully exempt from tax.</p>

For credit memos AutoInvoice defaults the tax exception id of the transaction you are crediting.

Validation: None

Destination: RA_CUSTOMER_TRX_LINES.TAX_EXEMPTION_ID

FOB_POINT

Enter the FOB point for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

For credit memos do not enter a value in this column. AutoInvoice uses the FOB point from the transaction you are crediting.

Validation: Must exist in AR_LOOKUPS.LOOKUP_CODE and AR_LOOKUPS.LOOKUP_TYPE = 'FOB'. Must be less than or equal to 30 characters in length

Destination: RA_CUSTOMER_TRX.FOB_POINT

GL_DATE

Enter the general ledger date for this transaction. The GL date determines the accounting period that you record this transaction to your general ledger. If the Post To GL option on the transaction type of the transaction being passed is set to No, the GL_DATE column should be NULL.

If LINE_TYPE = 'LINE', 'CHARGES', and you are passing transactions without rules or you are passing header freight, this column is optional.

If LINE_TYPE = 'LINE' and you are importing transactions with rules, do not enter a date in this column.

If LINE_TYPE = 'TAX' or 'FREIGHT' do not enter a value in this column.

For credit memos, AutoInvoice defaults to the date you run AutoInvoice, unless the transaction you are crediting is billed in arrears. In that case, AutoInvoice defaults to the GL date of the transaction you are crediting.

Please refer to the Deriving Dates section for a more details on general ledger date.

Validation: Must be in an open or future enterable accounting period and the period must exist in GL_PERIOD_STATUSES. If 'Post To GL' is set to No on the transaction type of the transaction being passed, column must be NULL.

Destination: RA_CUST_TRX_LINE_GL_DIST.GL_DATE

HEADER_ ATTRIBUTE1-15

Enter Descriptive Flexfield attribute information. Descriptive Flexfield attributes allow you to store additional columns, the contents of which you define.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional.

If LINE_TYPE = 'TAX' or 'FREIGHT', do not enter values in these columns.

Validation: None

Destination: RA_CUSTOMER_TRX.ATTRIBUTE1-15

HEADER_ ATTRIBUTE_ CATEGORY

Enter Descriptive Flexfield attribute category information which is shared between this transaction and other transactions. Descriptive Flexfield categories allow you to store different categories of attributes.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line', do not enter values in these columns.

Validation: None

Destination: RA_CUSTOMER_TRX.ATTRIBUTE_CATEGORY

INITIAL_ CUSTOMER_TRX_ID

This column is used by AutoInvoice and should be left null.

If this transaction is not a credit memo, AutoInvoice defaults a value into this column using RA_INTERFACE_LINES_ALL.REFERENCE_LINE_ID.

Validation: None

Destination: RA_CUSTOMER_TRX.INITIAL_CUSTOMER_TRX_ID

INTERFACE_LINE_ATTRIBUTE1-15

Enter the line Transaction Flexfield for this transaction. A Transaction Flexfield is a combination of attribute values which you use to uniquely identify this transaction in your original system. The reference value you enter here provides you with an audit trail from your Oracle Receivables application back to your original system. This is the primary key for RA_INTERFACE_LINES_ALL. You must enter values for enabled attributes.

Validation: Must not already exist together with INTERFACE_LINE_CONTEXT, in RA_CUSTOMER_TRX_LINES.

INTERFACE_LINE_ATTRIBUTE1-15, INTERFACE_LINE_CONTEXT and all enabled attributes must have values

Destination: RA_CUSTOMER_TRX.INTERFACE_HEADER_ATTRIBUTE1-15 and RA_CUSTOMER_TRX_LINES.INTERFACE_LINE_ATTRIBUTE1-15

INTERFACE_LINE_CONTEXT

This is a required column in AutoInvoice. Enter the context of the Line Transaction Flexfield entered in columns INTERFACE_LINE_ATTRIBUTE1-15. If you pass lines with global context set this column to 'Global Data Elements'.

Validation: None

Destination: RA_CUSTOMER_TRX_LINES.INTERFACE_LINE_CONTEXT

INTERFACE_LINE_ID

This column is used by AutoInvoice and should be left null. AutoInvoice defaults a value in this column using the RA_CUSTOMER_TRX_LINES_S sequence.

Validation: None

Destination: RA_CUSTOMER_TRX_LINES.CUSTOMER_TRX_LINE_ID

INTERFACE_STATUS

This column is used by AutoInvoice and should be left null. If AutoInvoice sets this column to 'P' then the line has been transferred successfully.

INTERNAL_NOTES

Enter internal notes for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing freight header, this column is optional.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter text in this column.

Validation: None

Destination: RA_CUSTOMER_TRX.INTERNAL_NOTES

INVENTORY_ ITEM_ID

Enter the inventory item id for this transaction.

If LINE_TYPE = 'LINE', or 'CHARGES' this column is optional. Depending on the value you entered for your batch source you can enter either a value in this column or a combination of segment values in MTL_SYSTEM_ITEMS_SEG1-20. If you specify segments in your batch source, AutoInvoice defaults a value in this column.

If LINE_TYPE = 'TAX' or 'FREIGHT', do not enter a value in this column.

For credit memos do not enter a value in this column. AutoInvoice uses the value from the transaction you are crediting.

Validation: Must exist in
MTL_SYSTEM_ITEMS.INVENTORY_ITEM_ID
and
MTL_SYSTEM_ITEMS.INVOICE_ENABLED_FLAG = 'Y'.

Destination: RA_CUSTOMER_TRX_LINES.INVENTORY_ITEM_ID

INVOICING_ RULE_ID

Enter the invoicing rule id for this transaction.

If LINE_TYPE = 'LINE' or you are passing header freight, this column is optional. For invoice lines with rules, you must enter either a value in this column or in INVOICING_RULE_NAME, depending on the value you entered for your batch source. If you specify invoicing rule name in your batch source, AutoInvoice defaults a value in this column.

If LINE_TYPE = 'TAX', 'CHARGES', or you are passing freight for a specific line, do not enter a value in this column.

For credit memos do not enter a value in this column. AutoInvoice uses the invoicing rule from the transaction you are crediting.

Validation: Must exist in RA_RULES.RULE_ID and
RA_RULES.RULE_ID = -2 or -3. If you enter an

INVOICING_ RULE_NAME

invoicing rule you must also enter an accounting rule. If LINE_TYPE = 'CHARGES' then this column must be null.

Destination: RA_CUSTOMER_TRX.INVOICING_RULE_ID

Enter the invoicing rule name for this transaction.

If LINE_TYPE = 'LINE' or you are passing header freight, this column is optional. For invoice lines with rules, you must enter either a value in this column or in INVOICING_RULE_ID, depending on the value you entered for your batch source. You can enter a value in this column or in INVOICE_RULE_ID.

If LINE_TYPE = 'TAX', 'CHARGES' or you are passing freight for a specific line, do not enter a value in this column.

For credit memos do not enter a value in this column. AutoInvoice uses the invoicing rule from the transaction you are crediting

Validation: Must exist in RA_RULES.RULE_ID and RA_RULES.RULE_ID = -2 or -3. If you enter an invoicing rule you must also enter an accounting rule. If LINE_TYPE = 'CHARGES' then this column must be null

Destination: None

LAST_PERIOD_ TO_CREDIT

For unit credit memos, enter the last period number from which you want to start crediting.

If this transaction is a credit memo against a transaction which uses an accounting rule and LINE_TYPE = 'LINE', CREDIT_METHOD_FOR_ACCT_RULE = 'UNIT', or you are passing header freight, you may enter a value in this column.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column. AutoInvoice will ignore any value that you enter in this column.

Validation: Must be between 0 and the invoice's accounting rule duration

Destination: RA_CUSTOMER_TRX_LINES.LAST_PERIOD_TO_CREDIT

LAST_UPDATE_LOGIN

This column is used by AutoInvoice and should be left null. AutoInvoice updates this column when it selects rows from the RA_INTERFACE_LINES_ALL table for processing.

Validation: None

Destination: None

LINE_NUMBER

This column is used by AutoInvoice and should be left null. AutoInvoice ignores any values passed in this column and always numbers the lines sequentially starting with the number 1 and in the order determined by the line ordering rule.

LINE_TYPE

Enter 'LINE', 'TAX', 'FREIGHT' or 'CHARGES' to specify the line type for this transaction. You must enter a value in this column.

For credit memos enter the type of line you are crediting.

Validation: Must be 'LINE', 'TAX', 'FREIGHT' or 'CHARGES'

Destination: RA_CUSTOMER_TRX_LINES.LINE_TYPE

LINK_TO_LINE_ATTRIBUTE1-15

Enter the link to your Transaction Flexfield attribute values.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, do not enter values in these columns.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, you must enter a value. Use link to line attributes to associate this tax or freight line to another transaction line in RA_INTERFACE_LINES_ALL. All tax lines and freight for specific lines must be associated to a line with a LINE_TYPE of 'LINE'. Enter the same combination of attribute values as the transaction to which you want to associate with.

For credit memos applied to tax lines, you must use these columns to link your credit memo tax lines to your credit memo transaction. Similarly, for credit memos applied to freight lines you must also use these columns to link your credit memo freight line to your credit memo transaction.

If you are applying a credit memo against a tax line which is linked to a transaction, you must enter a dummy credit memo transaction with a zero revenue amount and use these columns to link to your credit memo tax line. Similarly, if you are applying a credit memo against a freight line which is linked to a transaction, you must also enter a

dummy credit memo transaction with a zero revenue amount and use these columns to link to your credit memo freight line.

Validation: The transaction that you link to must have a LINE_TYPE = 'LINE'. You can only link at most one freight line to another transaction. You cannot link a transaction that has a LINE_TYPE = 'LINE' or 'CHARGES' to another transaction

Destination: None

LINK_TO_LINE_CONTEXT

Enter the context name of the Transaction Flexfield data that you entered in
RA_INTERFACE_LINES_ALL.LINK_TO_LINE_ATTRIBUTE1-15.

Validation:

Destination: None

LINK_TO_LINE_ID

This column is used by AutoInvoice and should be left null. AutoInvoice defaults a value into this column using
RA_INTERFACE_LINES_ALL.LINK_TO_LINE_ATTRIBUTE1-15 and
RA_INTERFACE_LINES_ALL.LINK_TO_LINE_CONTEXT.

Validation: None

Destination: RA_CUSTOMER_TRX_LINES.LINK_TO_CUST_TRX_LINE_ID

LOCATION_SEGMENT_ID

This column is used by AutoInvoice and should be left null. AutoInvoice defaults a value into this column if you are crediting a sales tax line.

Validation: None

Destination: RA_CUSTOMER_TRX_LINES.LOCATION_SEGMENT_ID

MEMO_LINE_ID

Enter the standard memo line id for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional. Depending on the value you entered for your batch source you can enter either a value in this column or in MEMO_LINE_NAME. If you specify memo line name in your batch source, AutoInvoice defaults a value in this column

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do

not enter a value in this column.

For credit memos do not enter a value in this column. AutoInvoice uses the memo line from the transaction you are crediting.

Validation: Must exist in AR_MEMO_LINES.MEMO_LINE_ID

Destination: RA_CUSTOMER_TRX_LINES.MEMO_LINE_ID

MEMO_LINE_NAME Enter the name of the standard memo line for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight this column is optional. Depending on the value you entered for your batch source you can enter either a value in this column or in MEMO_LINE_ID.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

For credit memos do not enter a value in this column. AutoInvoice uses the memo line from the transaction you are crediting

Validation: Must exist in AR_MEMO_LINES.NAME

Destination: None

MOVEMENT_ID This column is used to pass movement statistics that are tied to the shipment information and passed through AutoInvoice.

Autoinvoice will populate the column
RA_CUSTOMER_TRX_LINES.movement_id with
RA_INTERFACE_LINES_ALL movement_id and updates
MTL_MOVEMENT_STATISTICS with transaction information i.e.
customer_trx_id, batch_id, customer_trx_line_id.

Validation:

Destination: RA_CUSTOMER_TRX_LINES.movement_id

MTL_SYSTEM_ITEMS_SEG1-20 Assign a System Item Flexfield value for each segment you enable in your Oracle Receivables application. For example, if you enable six System Item Flexfield segments, you must enter six values in columns MTL_SYSTEM_ITEMS_SEG1-6. Be sure to enter the correct segment value. For example, value '01' is not the same as '1'.

If LINE_TYPE = 'LINE' or 'CHARGES', these columns are optional. Depending on the value you entered for your batch source you can enter either values in these columns or in INVENTORY_ITEM_ID.

If LINE_TYPE = 'TAX' or 'FREIGHT', do not enter values in these columns.

For credit memos do not enter values in these columns. AutoInvoice uses the values from the transaction you are crediting.

For debit memos do not enter values in these columns.

Validation: Valid combination of System Item Flexfield segment values

Destination: None

ORG_ID

If you use the Oracle Applications multiple organizations feature, enter the operating unit's org_id. If you do not use the Oracle Applications multiple organizations feature, leave this column null. To insert rows into open interface tables, use the multiple organization table RA_INTERFACE_LINES_ALL and explicitly populate the org_id column. You can determine an operating unit's org_id by querying the HR_OPERATING_UNITS view:

```
SELECT name, organization_id from  
HR_OPERATING_UNITS;
```

When you import rows *from* open interface tables, only rows from your current operating unit are processed.

ORIG_SYSTEM_BATCH_NAME

Enter the batch name for this transaction. This column is optional.

AutoInvoice does not perform any validation on this column but uses the value entered when grouping transactions into invoices.

Validation: None

Destination: RA_CUSTOMER_TRX.Orig_System_Batch_Name

ORIG_SYSTEM_BILL_ADDRESS_ID

Enter the Bill-To customer address id for this transaction. This Bill-To customer address id is for the Bill-To customer you entered in ORIG_SYSTEM_BILL_CUSTOMER_REF or ORIG_SYSTEM_BILL_CUSTOMER_ID.

If no default Remit-to Address has been specified, AutoInvoice uses the Bill-To address to determine the Remit-To address for the customer. If the Remit-To address cannot be determined, AutoInvoice will reject the invoice.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional. Depending on the value you entered for your batch source you must enter either a value in this column or in ORIG_SYSTEM_BILL_ADDRESS_REF. If you specify the Bill-To customer address reference in your batch source, AutoInvoice defaults a value in this column.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

Validation: RA_INTERFACE_LINES_ALL.Orig_System_Bill_Address_ID = RA_ADDRESSES.Address_ID
and
RA_INTERFACE_LINES_ALL.Orig_System_Bill_Customer_ID =
RA_CUSTOMERS.Customer_ID
and
RA_CUSTOMERS.Customer_ID =
RA_ADDRESSES.Customer_ID
and
RA_ADDRESSES.Address_ID =
RA_SITE_USES.Address_ID
and
RA_SITE_USES.Site_Use_Code = 'BILL_TO'

Destination: None

ORIG_SYSTEM_BILL_ADDRESS_REF Enter the Bill-To customer address reference from your original system. This reference is for the Bill-To customer you entered in ORIG_SYSTEM_BILL_CUSTOMER_REF or ORIG_SYSTEM_BILL_CUSTOMER_ID. The reference value you enter here provides you with an audit trail from Oracle Receivables and Oracle Government Receivables back to your original system.

If no default Remit-to Address has been specified, AutoInvoice uses the Bill-To address to determine the Remit-To address for the customer. If the Remit-To address cannot be determined, AutoInvoice will reject the invoice.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional. Depending on the value you entered for your batch source you must enter either a value in this column or in ORIG_SYSTEM_BILL_ADDRESS_ID.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

Validation: RA_INTERFACE_LINES_ALL.Orig_System_Bill_Address_Ref =
RA_ADDRESSES.Orig_System_Reference
and
RA_INTERFACE_LINES_ALL.Orig_System_Bill_Customer_Ref =
RA_CUSTOMERS.Orig_System_Reference
and
RA_CUSTOMERS.Customer_Id =
RA_ADDRESSES.Customer_Id
and
RA_ADDRESSES.Address_Id =
RA_SITE_USES.Address_Id
and
RA_SITE_USES.Site_Use_Code = 'BILL_TO'

Destination: None

Orig_System_Bill_Contact_Id

Enter the Bill-To contact id for this transaction. This Bill-To contact id must be for the Bill-To customer that you entered in Orig_System_Bill_Customer_Ref or Orig_System_Bill_Customer_Id.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional. Depending on the value you entered for your batch source you can enter either a value in this column or in Orig_System_Bill_Contact_Ref. If you specify the Bill-To customer contact reference in your batch source, AutoInvoice defaults a value in this column.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

Validation: RA_INTERFACE_LINES_ALL.Orig_System_Bill_Customer_Id =
RA_CONTACTS.Customer_Id
and
RA_INTERFACE_LINES_ALL.Orig_System_Bill_Contact_Id =

CONTACT_ID =
RA_CONTACTS.CONTACT_ID

Destination: RA_CUSTOMER_TRX.BILL_TO_CONTACT_ID

**ORIG_SYSTEM_
BILL_CONTACT_REF**

Enter the Bill-To contact reference from your original system. This reference is for the Bill-To customer that you entered in ORIG_SYSTEM_BILL_CUSTOMER_REF or ORIG_SYSTEM_BILL_CUSTOMER_ID. The reference value you enter here provides you with an audit trail from Oracle Receivables and Oracle Government Receivables back to your original system.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional. Depending on the value you entered for your batch source you can enter either a value in this column or in ORIG_SYSTEM_BILL_CONTACT_ID.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

Validation: RA_INTERFACE_LINES_ALL.Orig_System_Bill_Customer_ID =
RA_CONTACTS.CUSTOMER_ID
and
RA_INTERFACE_LINES_ALL.Orig_System_Bill_Customer_Ref =
RA_CONTACTS.Orig_System_Reference

Destination: None

**ORIG_SYSTEM_
BILL_CUSTOMER_ID**

Enter the Bill-To customer id for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional. Depending on the value you entered for your batch source you must enter either a value in this column or in ORIG_SYSTEM_BILL_CUSTOMER_REF. If you specify the Bill-To customer reference in your batch source, AutoInvoice defaults a value in this column.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

For credit memos you must enter the Bill-To customer id or the Bill-To customer id of a related customer of the transaction you are crediting.

Validation: Must exist in RA_CUSTOMERS.CUSTOMER_ID
Destination: RA_CUSTOMER_TRX.BILL_TO_CUSTOMER_ID

**ORIG_SYSTEM_
BILL_CUSTOMER_
REF**

Enter a value you can use to uniquely identify this Bill-To customer in your original system. The reference value you enter here provides you with an audit trail from Oracle Receivables and Oracle Government Receivables back to your original system.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional. Depending on the value you entered for your batch source you must enter either a value in this column or in ORIG_SYSTEM_BILL_CUSTOMER_ID.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

For credit memos you must enter the Bill-To customer reference or the Bill-To customer reference of a related customer of the transaction you are crediting.

Validation: Must exist in
RA_CUSTOMERS.ORIG_SYSTEM_REFERENCE

Destination: None

**ORIG_SYSTEM_
SHIP_ADDRESS_ID**

Enter the Ship-To customer address id for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional. Depending on the value you entered for your batch source you can enter either a value in this column or in ORIG_SYSTEM_SHIP_ADDRESS_REF. If you specify the Ship-To address reference in your batch source, AutoInvoice defaults a value in this column.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

For credit memos do not enter a value in this column, AutoInvoice uses the Ship-To address from the transaction you are crediting.

Validation: RA_INTERFACE_LINES_ALL.ORIG_SYSTEM_SHIP_
ADDRESS_ID = RA_ADDRESSES.ADDRESS_ID
and
RA_INTERFACE_LINES_ALL.ORIG_SYSTEM_SHIP_
ADDRESS_ID

CUSTOMER_ID =
 RA_CUSTOMERS.CUSTOMER_ID
 and
 RA_CUSTOMERS.CUSTOMER_ID =
 RA_ADDRESSES.CUSTOMER_ID
 and
 RA_ADDRESSES.ADDRESS_ID =
 RA_SITE_USES.ADDRESS_ID
 and
 RA_SITE_USES.SITE_USE_CODE = 'SHIP_TO'

Destination: None

ORIG_SYSTEM_SHIP_ADDRESS_REF

Enter a value you can use to uniquely identify this Ship-To customer address in your original system. The reference value you enter here provides you with an audit trail from your Oracle Receivables application back to your original system.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional. Depending on the value you entered for your batch source you can enter either a value in this column or in ORIG_SYSTEM_SHIP_ADDRESS_ID.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

For credit memos do not enter a value in this column, AutoInvoice uses the Ship-To address from the transaction you are crediting.

Validation:

RA_INTERFACE_LINES_ALL.Orig_System_Ship_Address_Ref =
 RA_ADDRESSES.Orig_System_Reference
 and
 RA_INTERFACE_LINES_ALL.Orig_System_Ship_Address_Ref =
 RA_CUSTOMERS.CUSTOMER_ID =
 RA_CUSTOMERS.CUSTOMER_ID =
 and
 RA_CUSTOMERS.CUSTOMER_ID =
 RA_ADDRESSES.CUSTOMER_ID
 and
 RA_ADDRESSES.ADDRESS_ID =
 RA_SITE_USES.ADDRESS_ID
 and
 RA_SITE_USES.SITE_USE_CODE = 'SHIP_TO'

Destination: None

**ORIG_SYSTEM_
SHIP_CONTACT_ID**

Enter the Ship-To contact id for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional. Depending on the value you entered for your batch source you can enter either a value in this column or in ORIG_SYSTEM_SHIP_CONTACT_REF. If you specify the Ship-To contact reference in your batch source, AutoInvoice defaults a value in this column.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

For credit memos, do not enter a value in this column. AutoInvoice uses the Ship-To contact from the transaction you are crediting.

Validation: RA_INTERFACE_LINES_ALL.Orig_System_Ship_Contact_Id = RA_CONTACTS.Customer_Id
and
RA_INTERFACE_LINES_ALL.Orig_System_Ship_Contact_Id = RA_CONTACTS.Contact_Id

Destination: RA_CUSTOMER_TRX.SHIP_TO_CONTACT_ID

**ORIG_SYSTEM_
SHIP_CONTACT_REF**

Enter a value you can use to uniquely identify this Ship-To contact in your original system. The reference value you enter here provides you with an audit trail from your Oracle Receivables application back to your original system

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional. Depending on the value you entered for your batch source you can enter either a value in this column or in ORIG_SYSTEM_SHIP_CONTACT_ID.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

For credit memos, do not enter a value in this column. AutoInvoice uses the Ship-To contact from the transaction you are crediting.

Validation: RA_INTERFACE_LINES_ALL.Orig_System_Ship_Contact_Ref = RA_CUSTOMER_TRX.Headers.Contact_Ref

CUSTOMER_ID =
RA_CONTACTS.CUSTOMER_ID
and
RA_INTERFACE_LINES_ALL.Orig_System_Ship_
IP_
CONTACT_REF =
RA_CONTACTS.Orig_System_Reference

Destination: None

Orig_System_Ship_Customer_ID

Enter the Ship-To customer id for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional. Depending on the value you entered for your batch source you can enter either a value in this column or in ORIG_SYSTEM_SHIP_CUSTOMER_REF. If you specify the Ship-To customer reference in your batch source, AutoInvoice defaults a value in this column.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

For credit memos, do not enter a value in this column. AutoInvoice uses the Ship-To customer from the transaction you are crediting.

Validation: Must exist in RA_CUSTOMERS.CUSTOMER_ID

Destination: RA_CUSTOMER_TRX.SHIP_TO_CUSTOMER_ID

Orig_System_Ship_Customer_Ref

Enter a value you can use to uniquely identify this Ship-To customer in your original system. The reference value you enter here provides you with an audit trail from your Oracle Receivables application back to your original system.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional. Depending on the value you entered for your batch source you can enter either a value in this column or in ORIG_SYSTEM_SHIP_CUSTOMER_ID.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

For credit memos, do not enter a value in this column. AutoInvoice uses the Ship-To customer from the transaction you are crediting.

Validation: Must exist in
RA_CUSTOMERS.Orig_System_Reference

	Destination:	None
ORIG_SYSTEM_SOLD_CUSTOMER_ID	Enter the Sold-To customer id for this transaction.	
	If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional. Depending on the value you entered for your batch source you can enter either a value in this column or in ORIG_SYSTEM_SOLD_CUSTOMER_REF. If you specify the Sold-To customer reference in your batch source, AutoInvoice defaults a value in this column.	
	If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.	
	For credit memos do not enter a value. AutoInvoice uses the Sold-To customer from the transaction you are crediting.	
	Validation:	Must exist in RA_CUSTOMERS.CUSTOMER_ID
	Destination:	RA_CUSTOMER_TRX.SOLD_TO_CUSTOMER_ID
ORIG_SYSTEM_SOLD_CUSTOMER_REF	Enter a value you can use to uniquely identify this Sold-To customer in your original system. The reference value you enter here provides you with an audit trail from your Oracle Receivables application back to your original system.	
	If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional. Depending on the value you entered for your batch source you can enter either a value in this column or in ORIG_SYSTEM_SOLD_CUSTOMER_ID.	
	If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.	
	For credit memos do not enter a value, AutoInvoice uses the Sold-To customer from the transaction you are crediting.	
	Validation:	Must exist in RA_CUSTOMERS.ORIG_SYSTEM_REFERENCE
	Destination:	None
PAYING_CUSTOMER_ID	This column is used by AutoInvoice and should be left null. Please refer to the section on Automatic Receipts for details on how AutoInvoice determines the paying customer.	
	Validation:	None

PAYING_CUSTOMER_ID	<p>Destination: RA_CUSTOMER_TRX.PAYING_CUSTOMER_ID</p> <p>This column is used by AutoInvoice and should be left null.</p> <p>Validation: None</p> <p>Destination: RA_CUSTOMER_TRX.PAYING_CUSTOMER_ID</p>
PAYING_SITE_USE_ID	<p>This column is used by AutoInvoice and should be left null.</p> <p>Validation: None</p> <p>Destination: RA_CUSTOMER_TRX.PAYING_SITE_USE_ID</p>
PAYING_SITE_USE_ID	<p>This column is used by AutoInvoice and should be left null. Please refer to the section on Automatic Receipts for details on how AutoInvoice determines the paying site use.</p> <p>Validation: None</p> <p>Destination: RA_CUSTOMER_TRX.PAYING_SITE_USE_ID</p>
PREVIOUS_CUSTOMER_TRX_ID	<p>This column is used by AutoInvoice and should be left null.</p> <p>For credit memos, AutoInvoice defaults a value into this column using RA_INTERFACE_LINES_ALL.REFERENCE_LINE_ID.</p> <p>Validation: None</p> <p>Destination: RA_CUSTOMER_TRX.PREVIOUS_CUSTOMER_TRX_ID and RA_CUSTOMER_TRX_LINES.CUSTOMER_TRX_ID</p>
PRIMARY_SALESREP_ID	<p>Enter the primary salesperson/agent id for this transaction.</p> <p>If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, and you entered Yes for the Require Salesreps/Agents system option, you must enter either a value in this column or in PRIMARY_SALESREP_NUMBER. Otherwise this column is optional. The value that you enter depends on the value you entered for your batch source. If you specify the primary salesrep/agent id in your batch source, AutoInvoice defaults a value in this column.</p> <p>If LINE_TYPE = "TAX" or you are passing freight for a specific line, do not enter a value in this column.</p> <p>Validation: Must exist in RA_SALESREPS.SALESREP_ID</p> <p>Destination: RA_CUSTOMER_TRX.PRIMARY_SALESREP_ID</p>

**PRIMARY_
SALESREP_NUMBER**

Enter the primary salesperson/agent number for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, and you entered Yes for the Require Salesreps/Agents system option, you must enter either a value in this column or in PRIMARY_SALESREP_ID. Otherwise this column is optional. The value that you enter depends on the value you entered for your batch source.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

Validation: Must exist in
RA_SALESREPS.SALESREP_NUMBER

Destination: None

PRINTING_OPTION

Enter the printing option for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional. AutoInvoice defaults to the printing option that you entered for this transaction type, if one was entered.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

Validation: Must exist in AR_LOOKUPS.LOOKUP_CODE and
AR_LOOKUP.LOOKUP_TYPE =
'INVOICE_PRINT_OPTIONS'

Destination: RA_CUSTOMER_TRX.PRINTING_OPTION

PURCHASE_ORDER

Enter the purchase order number for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

For credit memos do not enter a value in this column. AutoInvoice uses the purchase order number from the transaction you are crediting.

Validation: None

Destination: RA_CUSTOMER_TRX.PURCHASE_ORDER

PURCHASE_ ORDER_DATE

Enter the date of the purchase order for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

For credit memos, do not enter a value in this column. AutoInvoice uses the purchase order date from the transaction you are crediting.

Validation: None

Destination: RA_CUSTOMER_TRX.PURCHASE_ORDER_DATE

PURCHASE_ ORDER_REVISION

Enter the purchase order revision for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

For credit memos, do not enter a value in this column. AutoInvoice uses the purchase order revision from the transaction you are crediting.

Validation: None

Destination: RA_CUSTOMER_TRX.PURCHASE_ORDER_REVISION

QUANTITY

If this transaction is an invoice or credit memo line and LINE_TYPE = 'LINE' or you are passing header freight, this column is optional. For invoice lines, enter the number of units shipped. For credit memo lines, enter the number of units you are crediting. If you do not enter a value in this column, AutoInvoice uses AMOUNT as the extended amount for this transaction. If this transaction is a dummy line for either freight only or tax only, AutoInvoice ignores the value you enter in this column.

Exception: If this is a Credit Memo line and LINE_TYPE = 'LINE', CREDIT_METHOD_FOR.ACCT_RULE = 'UNIT' then this column is mandatory.

For Debit Memos, if LINE_TYPE = 'CHARGES', set quantity to 1.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

For Credit Memos, if LINE_TYPE = 'CHARGES', set quantity to 1 or -1.

Validation: For Debit Memos lines with LINE_TYPE = 'CHARGES', quantity must be 1. For Credit Memo lines with LINE_TYPE = 'CHARGES', this column must be 1 or -1.

For Credit Memo lines with LINE_TYPE = 'LINE' and CREDIT_METHOD_FOR_ACCT_RULE = 'UNIT' then this column must be not null.

Destination: RA_CUSTOMER_TRX_LINES.QUANTITY_INVOICED if this transaction is an invoice line.
RA_CUSTOMER_TRX_LINES.QUANTITY_CREDITED if this transaction is a credit memo line

QUANTITY_ ORDERED

Enter the original number of units ordered for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing freight for a specific line, this column is optional.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this field.

For credit memos, do not enter a value in this column. AutoInvoice uses the quantity ordered from the transaction you are crediting.

Validation: None

Destination: RA_CUSTOMER_TRX_LINES.QUANTITY_ORDERED

REASON_CODE

Enter the reason code for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional. Depending on the value you entered for your batch source you can enter either a value in this column or in REASON_CODE_MEANING. If you specify the reason code meaning in your batch source, AutoInvoice defaults a value in this column.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

For credit memos and on-account credits this column is optional.

Validation: Must exist in AR_LOOKUPS.LOOKUP_CODE.
This lookup type is either INVOICING_REASON
or CREDIT_MEMO_REASON

Destination: RA_CUSTOMER_TRX_LINES.REASON_CODE
and RA_CUSTOMER_TRX.REASON_CODE

REASON_CODE_ MEANING

Enter the meaning of the reason code for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional. Depending on the value you entered for your batch source you can enter either a value in this column or in REASON_CODE.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

For credit memos and on-account credits this column is optional.

Validation: Must exist in AR_LOOKUPS.MEANING. This
lookup type is either INVOICING_REASON or
CREDIT_MEMO_REASON

Destination: None

RECEIPT_METHOD_ ID

Enter the payment method id for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional. Depending on the value you entered for your batch source you can enter either a value in this column or in RECEIPT_METHOD_NAME. If you specify the payment method name in your batch source, AutoInvoice defaults a value in this column. AutoInvoice always defaults the payment method using the following hierarchy:

1. primary receipt method of the parent primary bill-to site
2. primary receipt method of the parent customer
3. primary receipt method of the bill-to site
4. primary receipt method of the bill-to customer

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this field.

Validation: Must exist in
AR_RECEIPT_METHODS.RECEIPT_METHOD_

ID and must belong to the bill-to customer or the parent. Additionally, the payment method must have at least one bank account in the same currency as the transaction or have its' Receipts Multi-Currency flag set to Yes.

Destination: RA_CUSTOMER_TRX.RECEIPT_METHOD_ID

**RECEIPT_METHOD_
NAME**

Enter the name of the payment method for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional. Depending on the value you entered for your batch source you can enter either a value in this column or in RECEIPT_METHOD_ID.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this field.

Validation: Must exist in AR_RECEIPT_METHODS.NAME and must belong to the bill-to customer or the parent.

Destination: None

**REFERENCE_LINE_
ATTRIBUTE1-15**

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, and this transaction is a credit memo, you must enter either the Transaction Flexfield of the transaction line you are crediting in these columns or the RA_CUSTOMER_TRX_LINES.CUSTOMER_TRX_LINE_ID of the transaction you are crediting in RA_INTERFACE_LINES_ALL.REFERENCE_LINE_ID. Otherwise, do not enter values in these columns.

If LINE_TYPE = 'TAX' and this transaction is a credit memo, you must enter either the Transaction Flexfield of the tax line you are crediting in these columns or the RA_CUSTOMER_TRX_LINES.CUSTOMER_TRX_LINE_ID of the transaction tax line you are crediting in RA_INTERFACE_LINES_ALL.REFERENCE_LINE_ID. Otherwise, do not enter values in these columns.

If LINE_TYPE= 'FREIGHT' and this transaction is a credit memo, you must enter either the Transaction Flexfield of the freight line you are crediting in these columns or the RA_CUSTOMER_TRX_LINES.CUSTOMER_TRX_LINE_ID of the transaction freight line you are crediting in RA_INTERFACE_LINES_ALL.REFERENCE_LINE_ID. Otherwise, do

not enter values in these columns.

For on-account credits do not enter values in these columns.

Validation: Must exist in
RA_CUSTOMER_TRX_LINES.INTERFACE_LINE_
ATTRIBUTE1-15 or
RA_INTERFACE_LINES_ALL.INTERFACE_LINE_
ATTRIBUTE1-15

Destination: None

REFERENCE_LINE_ CONTEXT

Enter the context name of the Transaction Flexfield data entered in
RA_INTERFACE_LINES_ALL.REFERENCE_LINE_A TTRIBUTE1-15.
You must enter a value in this column if you entered values in
RA_INTERFACE_LINES_ALL.A TTRIBUTE1-15.

Validation: Must exist in:
RA_CUSTOMER_TRX_LINES.INTERFACE_LINE_
CONTEXT
or
RA_INTERFACE_LINES_ALL.INTERFACE_LINE_
CONTEXT

Destination: None

REFERENCE_LINE_ ID

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header
freight, and this transaction is a credit memo, you must enter the
RA_CUSTOMER_TRX_LINES.CUSTOMER_TRX_LINE_ID of the
transaction line you are crediting in this column or the Transaction
Flexfield in REFERENCE_LINE_ATTRIBUTE1-15. Otherwise, do not
enter a value.

If LINE_TYPE = 'LINE' and this transaction is an invoice against a
commitment, you must enter the RA_CUSTOMER_TRX_LINES.
CUSTOMER_TRX_LINE_ID of the commitment line you are
referencing.

If LINE_TYPE= 'TAX' and this transaction is a credit memo, you must
enter the RA_CUSTOMER_TRX_LINES.CUSTOMER_TRX_LINE_ID of
the tax line you are crediting in these columns or the Transaction
Flexfield in REFERENCE_LINE_ATTRIBUTE1-15. Otherwise, do not
enter a value in this column.

If LINE_TYPE = 'FREIGHT' and this transaction is a credit memo, you

must enter the
RA_CUSTOMER_TRX_LINES.CUSTOMER_TRX_LINE_ID of the
freight line you are crediting in these columns or the Transaction
Flexfield in REFERENCE_LINE_ATTRIBUTE1-15. Otherwise, do not
enter a value in this column.

For on-account credits do not enter a value in this column.

Validation: Must exist in
RA_CUSTOMER_TRX_LINES.CUSTOMER_TRX_
LINE_ID

Destination: RA_CUSTOMER_TRX_LINES.PREVIOUS_
CUSTOMER_TRX_LINE_ID if this transaction is a
credit memo. Otherwise, RA_CUSTOMER_TRX_
LINES.INITIAL_CUSTOMER_TRX_LINE_ID

RELATED_BATCH_ SOURCE_NAME

Enter the name of the batch source of the document to which this
transaction is related.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header
freight, this column is optional. Depending on the value you entered
for your batch source you can enter a value in this column and the
related transaction number in RELATED_TRX_NUMBER. Or, you can
enter the related customer transaction id in
RELATED_CUSTOMER_TRX_ID.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do
not enter a value in this column.

For credit memos and on-account credits do not enter a value in this
column.

Validation: RA_INTERFACE_LINES_ALL.RELATED_BATCH_
—
SOURCE_NAME =
RA_BATCH_SOURCES.NAME
and
RA_INTERFACE_LINES_ALL.RELATED_TRX_
NUMBER = RA_CUSTOMER_TRX.TRX_NUMBER
and
RA_BATCH_SOURCES.BATCH_SOURCE_ID =
RA_CUSTOMER_TRX.BATCH_SOURCE_ID

Destination: None

RELATED_ CUSTOMER_TRX_ ID

Enter the customer transaction id of the document to which this transaction is related.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional. Depending on the value you entered for your batch source you can enter a value in this column. Or, you can enter the related transaction number in RELATED_TRX_NUMBER and the related batch source name in RELATED_BATCH_SOURCE_NAME.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

For credit memos and on-account credits do not enter a value in this column.

Validation: Must exist in
RA_CUSTOMER_TRX.CUSTOMER_TRX_ID

Destination: RA_CUSTOMER_TRX.RELATED_CUSTOMER_
TRX_ID

RELATED_TRX_ NUMBER

Enter the document number to which this transaction is related.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional. Depending on the value you entered for your batch source you can enter a value in this column and the related batch source name in RELATED_BATCH_SOURCE_NAME. Or, you can enter the related customer transaction id in RELATED_CUSTOMER_TRX_ID.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

For credit memos and on-account credits do not enter a value in this column.

Validation: RA_INTERFACE_LINES_ALL.RELATED_BATCH
-
SOURCE_NAME =
RA_BATCH_SOURCES.NAME
and
RA_INTERFACE_LINES_ALL.RELATED_TRX_
NUMBER = RA_CUSTOMER_TRX.TRX_NUMBER
and
RA_BATCH_SOURCES.BATCH_SOURCE_ID =
RA_CUSTOMER_TRX.BATCH_SOURCE_ID

	Destination:	None
REQUEST_ID	This column is used by AutoInvoice and should be left null.	
	Validation:	None
	Destination:	The REQUEST_ID column in RA_CUSTOMER_TRX, RA_CUSTOMER_TRX_LINES, RA_CUST_TRX_LINE_GL_DIST, AR_PAYMENT_SCHEDULES, AR_RECEIVABLE_APPLICATIONS, AR_ADJUSTMENTS and RA_CUST_TRX_LINE_SALESREPS
RULE_START_DATE	Enter the date that you want to start the accounting rule for this transaction.	
	If LINE_TYPE = 'LINE' or you are passing header freight, this column is optional. If you specify Specific Date in your accounting rule do not enter a value in this column.	
	If LINE_TYPE = 'TAX', 'CHARGES', or you are passing freight for a specific line, do not enter a value in this column.	
	For credit memos do not enter a value in this column.	
	Please refer to the Deriving Dates section for a more details on rule start date.	
	Validation:	None
	Destination:	RA_CUSTOMER_TRX_LINES.RULE_START_DATE
SALES_ORDER	Enter the sales order number for this transaction.	
	If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional.	
	If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.	
	For credit memos do not enter a value in this column. AutoInvoice uses the sales order number from the transaction you are crediting.	
	Validation:	None

Destination: RA_CUSTOMER_TRX__LINES.SALES_ORDER

**SALES_ORDER_
DATE**

Enter the date of the sales order for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

For credit memos do not enter a value in this column. AutoInvoice uses the sales order date from the transaction you are crediting.

Enter the date of the revenue order for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

For credit memos do not enter a value in this column. AutoInvoice uses the revenue order date from the transaction you are crediting.

Validation: None

Destination: RA_CUSTOMER_TRX__LINES.SALES_ORDER_
DATE

SALES_ORDER_ LINE Enter the sales order line number for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

For credit memos do not enter a value in this column. AutoInvoice uses the sales order line number from the transaction you are crediting.

Validation: None

Destination: RA_CUSTOMER_TRX__LINES.SALES_ORDER_
LINE

**SALES_ORDER_
REVISION**

Enter the sales order revision for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header

freight, this column is optional.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

For credit memos do not enter a value in this column. AutoInvoice uses the sales order revision from the transaction you are crediting.

Validation: None

Destination: RA_CUSTOMER_TRX_LINES.SALES_ORDER_REVISION

SALES_ORDER_SOURCE

Enter the source of the sales order for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

For credit memos do not enter a value in this column. AutoInvoice uses the source of the sales order from the transaction you are crediting.

Validation: None

Destination: RA_CUSTOMER_TRX_LINES.SALES_ORDER_SOURCE

SALES_TAX_ID

This column is used by AutoInvoice and should be left null.

For credit memos, AutoInvoice defaults to the sales tax id of the transaction you are crediting.

Validation: None

Destination: RA_CUSTOMER_TRX_LINES.SALES_TAX_ID

SET_OF_BOOKS_ID

Enter the set of books id for this transaction. You must enter a value in this column.

Validation: Must exist in
AR_SYSTEM_PARAMETERS.SET_OF_BOOKS_ID

Destination: RA_CUSTOMER_TRX.SET_OF_BOOKS_ID

SHIP_DATE_ACTUAL

Enter the shipment date for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

For credit memos do not enter a value in this column. AutoInvoice uses the shipment date from the transaction you are crediting.

Validation: None

Destination: RA_CUSTOMER_TRX.SHIP_DATE_ACTUAL

SHIP_VIA

Enter the ship via code for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

For credit memos do not enter a value in this column. AutoInvoice uses the ship via code from the transaction you are crediting.

Validation: ORG_FREIGHT_CODE =
RA_INTERFACE_LINES_ALL.SHIP_VIA and
ORG_FREIGHT.ORGANIZATION_ID = (Your
Organization ID)
RA_INTERFACE_LINES_ALL.SHIP_VIA – must
be less than or equal to 25 characters in length

Destination: RA_CUSTOMER_TRX.SHIP_VIA

TAX_CODE

Enter the tax code for this tax line.

If LINE_TYPE = 'LINE', 'CHARGES', or 'FREIGHT', do not enter a value in this column.

If LINE_TYPE = 'TAX', this column is mandatory.

For credit memos AutoInvoice defaults the tax code from the transaction you are crediting.

Validation: Must exist in AR_VAT_TAX.TAX_CODE

Destination: None

TAX_EXEMPT_FLAG

If LINE_TYPE = 'LINE', this column is optional. The value you enter here controls how a line is taxed.

Enter 'E' if you want AutoInvoice to exempt an invoice line that would normally be taxed and your system option 'Use Customer Exemptions' is set to Yes. If you enter 'E' you must enter a value for TAX_EXEMPT_REASON_CODE or TAX_EXEMPT_REASON_CODE_MEANING depending on your batch source option.

Enter 'R' if you want AutoInvoice to force tax on an invoice line, ignoring any exemption certificates that may be on file.

Enter 'S' if you want tax to be calculated as per the normal procedures set up in your Oracle Receivables application

For all other line types, do not enter a value in this column.

For credit memos do not enter a value in this column.

Validation: Must exist in AR_LOOKUPS.LOOKUP.CODE
Lookup type is TAX_CONTROL_FLAG

Destination: RA_CUSTOMER_TRX_LINES.
TAX_EXEMPT_FLAG

TAX_EXEMPT_ NUMBER

Enter the tax exempt number for this transaction.

If LINE_TYPE = 'LINE' and tax_exempt_flag = 'E', then you may enter a value in this column.

Otherwise, do not enter a value in this column.

For all other line types, do not enter a value in this column.

For credit memos do not enter a value in this column.

Validation: None

Destination: RA_CUSTOMER_TRX_LINES.
TAX_EXEMPT_NUMBER

TAX_EXEMPT_ REASON_CODE

Enter the tax exempt reason code for this transaction. If LINE_TYPE = 'LINE' and tax_exempt_flag = 'E', then depending on your batch source option, Memo Reason, you must enter a value in this column or in TAX_EXEMPT_REASON_CODE_MEANING. If you specify a tax exempt reason code meaning in your batch source, AutoInvoice defaults it in this column.

For all other line types, do not enter a value in this column.

For credit memos do not enter a value in this column.

Validation: Must exist in AR_LOOKUPS.LOOKUP.CODE
Lookup type is TAX_REASON

Destination: RA_CUSTOMER_TRX_LINES.
TAX_EXEMPT_REASON_CODE

TAX_EXEMPT_ RAESON_CODE _MEANING

Enter the tax exempt reason code meaning for this transaction. If LINE_TYPE = 'LINE' and tax_exempt_flag = 'E', then depending on your batch source option, Memo Reason, you must enter a value in this column or in TAX_EXEMPT_REASON_CODE.

Otherwise, do not enter a value in this column.

For all other line types, do not enter a value in this column.

For credit memos do not enter a value in this column.

Validation: Must exist in AR_LOOKUPS.MEANING. Lookup type is TAX_REASON

Destination: None

TAX_PRECEDENCE

Enter the precedence number for this tax line. This column is used to compute tax compounding.

If LINE_TYPE = 'LINE', 'CHARGES', or 'FREIGHT', do not enter a value in this column.

If LINE_TYPE = 'TAX' and you allow compound tax, you can enter a value in this column. Otherwise do not enter a value.

If you are passing freight for a specific line, do not enter a value in this column.

For credit memos AutoInvoice defaults the tax precedence from the transaction you are crediting.

Validation: None

Destination: RA_CUSTOMER_TRX_LINES.TAX_
PRECEDENCE

TAX_RATE

Enter the tax rate for this tax line.

If LINE_TYPE = 'LINE', 'CHARGES', or 'FREIGHT', do not enter a value in this column.

If LINE_TYPE = 'TAX', you must enter a value either in this column or the AMOUNT column. Any exemptions for the tax lines must be

factored into the tax rate.

Validation: None

Destination: RA_CUSTOMER_TRX_LINES.TAX_RATE

TERM_ID

Enter the payment term id for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional. Depending on the value you entered for your batch source you must enter either a value in this column or in TERM_NAME. If you specify term name in your batch source, AutoInvoice defaults a value in this column.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

For credit memos and on-account credits do not enter a value in this column.

Validation: Must exist in RA_TERMS.TERM_ID

Destination: RA_CUSTOMER_TRX.TERM_ID

TERM_NAME

Enter the name of the payment term for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional. Depending on the value you entered for your batch source you must enter either a value in this column or in TERM_ID.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

For credit memos and on-account credits do not enter a value in this column.

Validation: Must exist in RA_TERMS.NAME

Destination: None

TERRITORY_ID

Enter the territory id for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional. Depending on the value you entered for your batch source you can enter either a value in this column or a combination of territory segment values in

TERRITORY_SEGMENT1-20. If you specify the combination of territory segment values in your batch source, AutoInvoice defaults a value in this column.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

For credit memos do not enter a value in this column. AutoInvoice uses the territory from the transaction you are crediting.

Validation: Must exist in RA_TERRITORIES.TERRITORY_ID

Destination: RA_CUSTOMER_TRX.TERRITORY_ID

TERRITORY_ SEGMENT1-20

Assign a Territory Flexfield value for each segment you enable in your Oracle Receivables application. For example, if you enable six Territory Flexfield segments, you must enter six values in columns TERRITORY_SEGMENT1-6. Be sure to enter the correct segment value. For example, value '01' is not the same as '1'.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, these columns are optional. Depending on the value you entered for your batch source you can enter either values in these columns or in TERRITORY_ID.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter values in these columns.

For credit memos do not enter values in these columns. AutoInvoice uses the territory from the transaction you are crediting.

Validation: Valid combination of Territory Flexfield segment values from RA_TERRITORIES

Destination: None

TRX_DATE

Enter the transaction date for this transaction.

If TRX_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional. If this transaction is an invoice or debit memo line, you can enter the invoice date. If this transaction is a credit memo line, you can enter the credit memo date. If this transaction is an invoice line and uses an Arrears Invoice invoicing rule, do not enter a value in this column.

If you do not enter a transaction date, AutoInvoice uses the general ledger date for invoice and debit memo lines. For credit memo lines,

AutoInvoice uses the following hierarchy: credit memo general ledger date, and the general ledger date for the invoice's receivable distribution or the date on the Run AutoInvoice form, whichever is later.

When child invoices are created against a commitment, AutoInvoice ensures that the child invoice's transaction date falls between the commitment's start and end dates.

If TRX_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

Validation: None

Destination: RA_CUSTOMER_TRX.TRX_DATE

TRX_NUMBER

Enter the number for this transaction.

If TRX_TYPE = 'LINE', 'CHARGES', or you are passing header freight, and your batch source has Automatic Invoice Numbering set to No, you must enter a value in this column.

If TRX_TYPE = 'LINE', 'CHARGES', or you are passing header freight, and your batch source has Automatic Invoice Numbering set to Yes, do not enter a value in this column. AutoInvoice inserts a unique number in this column.

If TRX_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

Validation: Must not already exist in
RA_CUSTOMER_TRX.TRX_NUMBER
and RA_CUSTOMER_TRX.BATCH_SOURCE_ID

Destination: RA_CUSTOMER_TRX.TRX_NUMBER and
AR_PAYMENT_SCHEDULES.TRX_NUMBER

UOM_CODE

Enter the unit of measure code for this transaction.

If LINE_TYPE = 'LINE' and the line has an item you must enter either a value in this column or in UOM_NAME. If this a freight-only line, a tax-only line, or a line with no item, this column is optional.

If LINE_TYPE = 'LINE' and you are passing a dummy line for either a tax-only or freight-only line, AutoInvoice ignores what you enter here.

If LINE_TYPE = 'TAX', 'CHARGES', or you are passing freight for a

specific line, do not enter a value in this column.
For credit memos do not enter a value in this column. AutoInvoice uses the unit of measure from the transaction you are crediting.

Validation: Must exist in
MTL_UNITS_OF_MEASURE.UOM_CODE If
Line_type = 'CHARGES', then this column must be
null.

Destination: RA_CUSTOMER_TRX_LINES.UOM_CODE

UOM_NAME

Enter the unit of measure name for this transaction.

If LINE_TYPE = 'LINE' and the line has an item you must enter either a value in this column or in UOM_CODE. If this a freight-only line, a tax-only line, or a line with no item, this column is optional.

If LINE_TYPE = 'LINE' or you are passing header freight, and you are passing a dummy line for either a tax-only or freight-only line, AutoInvoice ignores what you enter here.

If LINE_TYPE = 'TAX', 'CHARGES', or you are passing freight for a specific line, do not enter a value in this column.

For credit memos do not enter a value in this column. AutoInvoice uses the unit of measure from the transaction you are crediting.

Validation: Must exist in
MTL_UNITS_OF_MEASURE.UNIT_OF_
MEASURE. If LINE_TYPE = 'CHARGES' then this
column must be null.

Destination: None

UNIT_SELLING_ PRICE

Enter the selling price per unit for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional. If you do not enter a value in this column, AutoInvoice defaults to the amount in RA_INTERFACE_LINES_ALL.AMOUNT as the extended amount for this transaction.

If LINE_TYPE = 'LINE' or you are passing header freight, and you are passing a dummy line for either a tax-only or freight-only line, AutoInvoice ignores what you enter here.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

Validation: None

Destination: RA_CUSTOMER_TRX_LINES.UNIT_SELLING_PRICE

UNIT_STANDARD_PRICE

Enter the standard price per unit for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

For credit memos, do not enter a value in this column. AutoInvoice uses the unit standard price from the transaction you are crediting.

Validation: None

Destination: RA_CUSTOMER_TRX_LINES.UNIT_STANDARD_PRICE

USSGL_TRANSACTION_CODE

Enter the transaction code for this transaction. If this transaction is linked to another transaction, you must enter the same transaction code as the one to which it is linked. This column is optional.

Validation: Must be a valid transaction code as defined in your value set for this flexfield.

Destination: RA_CUSTOMER_TRX_LINES.USSGL_TRANSACTION_CODE

USSGL_TRANSACTION_CODE_CONTEXT

This column is not currently used by AutoInvoice.

Validation: None

Destination: None

VAT_TAX_ID

This column is used by AutoInvoice and should be left null. If you enter a value in TAX_CODE, AutoInvoice defaults a value in this column.

For credit memos AutoInvoice defaults to the VAT tax id of the transaction you are crediting.

Validation: None

Destination: RA_CUSTOMER_TRX_LINES.VAT_TAX_ID

WAYBILL_NUMBER

Enter the waybill number for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, this column is optional.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line do not enter a value in this column.

For credit memos do not enter a value in this column. AutoInvoice uses the waybill number from the transaction you are crediting.

Validation: None

Destination: RA_CUSTOMER_TRX.WAYBILL_NUMBER

RA_INTERFACE_SALESCREDITS_ALL

This table is used to import sales or revenue credit information for your transactions. This table must be populated if your autoaccounting is set up to derive segment values based on the salesrep. If autoaccounting does not depend on salesrep then the value you enter in the Require Salesrep/Agent field of the Define System Options form and Allow Sales/Revenue Credits field in the Define Invoice Sources field will determine whether you must enter sales or revenue credit information.

If you are importing invoices, debit memos and on account credits and your system option requires salesperson or agent, you must provide sales or revenue credit information, regardless of the value entered in the Allow Sales/Revenue Credit field for your invoice batch source.

If you are importing credit memos and your system option requires that you enter a salesperson or agent, you can provide sales or revenue credit information. If you do not provide sales or revenue credit information, AutoInvoice uses sales or revenue credit information from the invoice you are crediting. If the invoice you are crediting does not have sales or revenue credit information, AutoInvoice creates a 100% 'No Sales Credit' or 'No Revenue Credit' line for this invoice. This sales or revenue credit line is then used to determine the sales or revenue credit amount for the credit memo.

Regardless of the type of transaction you are importing, if your system option does not require salesperson or agent, but your invoice batch source allows sales or revenue credits, you can provide sales or

revenue credit information. AutoInvoice will validate it and pass this information with your transaction. If your system option does not require salesperson or agent and your invoice batch source does not allow sales or revenue credits, do not provide sales or revenue credit information. AutoInvoice ignores any values that you pass.

ATTRIBUTE1-15

Enter the Descriptive Flexfield attribute information for this sales or revenue credit assignment. Descriptive Flexfield attributes allow you to store additional columns, the contents of which you define. These columns are optional.

Validation: None

Destination: RA_CUST_TRX_LINES_SALESREPS.
ATTRIBUTE1-15

ATTRIBUTE_ CATEGORY

Enter the Descriptive Flexfield category information for this sales credit assignment. Descriptive Flexfield categories allow you to store different categories of attributes. This column is optional.

Validation: None

Destination: RA_CUST_TRX_LINE_SALESREPS.ATTRIBUTE_
CATEGORY

INTERFACE_LINE_ ATTRIBUTE1-15

Enter the same Transaction Flexfield for the transaction with which you want to associate this sales or revenue credit assignment. The values you enter here provide you with an audit trail from your Oracle Receivables application back to your original system. You must enter a value for each attribute you enabled.

INTERFACE_LINE_ CONTEXT

Enter the context name of the Transaction Flexfield data that you entered in RA_INTERFACE_SALESCREDITS_ALL.
INTERFACE_LINE_ATTRIBUTE1-15. You must enter a value.

Validation: None

Destination: None

INTERFACE_LINE_ID

This column is used by AutoInvoice and should be left null. AutoInvoice defaults a value into this column using
RA_INTERFACE_SALESCREDITS_ALL.INTERFACE_LINE_
ATTRIBUTE1-15.

Validation: None

Destination: RA_CUST_TRX_LINE_SALESREPS.CUSTOMER_
TRX_LINE_ID

INTERFACE_ SALESCREDIT_ID	<p>This column is used by AutoInvoice and should be left null. AutoInvoice defaults a value into this column using the sequence RA_CUST_TRX_LINE_SALESREPS_S.</p> <p>Validation: None</p> <p>Destination: RA_CUST_TRX_LINE_SALESREPS.CUST_TRX_LINE_SALESREP_ID</p> <p>Validation: None</p> <p>Destination: None</p>
INTERFACE_STATUS	<p>This column is used by AutoInvoice and should be left null.</p> <p>Validation: None</p> <p>Destination: None</p>
LAST_UPDATE_ LOGIN	<p>This column is used by AutoInvoice and should be left null. AutoInvoice updates this column when it selects rows from the RA_INTERFACE_SALESCREDITS_ALL table for processing.</p> <p>Validation: None</p> <p>Destination: None</p>
ORG_ID	<p>If you use the Oracle Applications multiple organizations feature, enter the operating unit's org_id. If you do not use the Oracle Applications multiple organizations feature, leave this column null. To insert rows into open interface tables, use the multiple organization table RA_INTERFACE_SALESCREDITS_ALL and explicitly populate the org_id column. You can determine an operating unit's org_id by querying the HR_OPERATING_UNITS view:</p> <pre>SELECT name, organization_id from HR_OPERATING_UNITS;</pre> <p>When you import rows <i>from</i> open interface tables, only rows from your current operating unit are processed.</p>
REQUEST_ID	<p>This column is used by AutoInvoice and should be left null.</p> <p>Validation: None</p> <p>Destination: None</p>
SALES_CREDIT_ AMOUNT_SPLIT	<p>Enter the sales or revenue credit amount for this salesperson or agent. This column is optional. Depending on the value you entered for your batch source you must enter either a value in this column or in SALES_CREDIT_PERCENT_SPLIT. If you specify the sales or revenue</p>

credit percent in your batch source, AutoInvoice defaults a value in this column.

Validation: If the sales or revenue credit for this sales/revenue credit assignment is of type Quota, the sum of sales/revenue credit amounts for a transaction must equal the amount of the transaction.

Destination: RA_CUST_TRX_LINE_SALESREPS.
REVENUE_AMOUNT_SPLIT if the sales credit type is Quota.
RA_CUST_TRX_LINE_SALESREPS.
NON_REVENUE_AMOUNT_SPLIT if the sales credit type is not Quota

SALES_CREDIT_ PERCENT_SPLIT

Enter the sales or revenue credit percent for this salesperson/agent. This column is optional. Depending on the value you entered for your batch source you must enter either a value in this column or in SALES_CREDIT_AMOUNT_SPLIT. If you specify the sales or revenue credit amount in your batch source, AutoInvoice defaults a value in this column.

Validation: Your sales or revenue credit percent must be between 0 and 100, and if sales/revenue credit type is Quota, the sales or revenue credit percentage for a transaction must sum to 100.

SALES_CREDIT_ TYPE_ID

Enter the id of the sales or revenue credit type for this sales/revenue credit assignment. This column is optional. Depending on the value you entered for your batch source you must enter either a value in this column or in SALES_CREDIT_TYPE_NAME. If you specify the sales/revenue credit type name in your batch source, AutoInvoice defaults a value in this column.

Validation: Must exist in
SO_SALES_CREDIT_TYPES.SALES_CREDIT_
TYPE_ID

Destination: None

SALES_CREDIT_ TYPE_NAME

Enter the name of the sales/revenue credit type for this sales/revenue credit assignment. This column is optional. Depending on the value you entered for your batch source you must enter either a value in this column or in SALES_CREDIT_TYPE_ID.

Validation: Must exist in SO_SALES_CREDIT_TYPES.NAME

Destination: None

SALESREP_ID

Enter the salesperson/agent id for this sales/revenue credit assignment. This column is optional. Depending on the value you entered for your batch source you must enter either a value in this column or in SALESREP_NUMBER. If you specify the salesperson/agent number in your batch source, AutoInvoice defaults a value in this column.

Validation: Must exist in RA_SALESREPS.SALESREP_ID

Destination: RA_CUST_TRX_LINE_SALESREPS.SALESREP_ID

SALESREP_NUMBER

Enter the salesperson/agent number for this sales/revenue credit assignment. This column is optional. Depending on the value you entered for your batch source you must enter either a value in this column or in SALESREP_ID.

Validation: Must exist in
RA_SALESREPS.SALESREP_NUMBER

Destination: None

RA_INTERFACE_DISTRIBUTIONS_ALL

If you do not use AutoAccounting you must enter accounting distributions for your transactions. Otherwise, AutoInvoice does not require you to enter accounting distributions for your transactions.

If your accounting distributions are for transactions that use accounting rules, you must enter the percentages, but not the amounts. If you enter the amounts, AutoInvoice will ignore those values.

If your accounting distributions are for transactions that do not use accounting rules, you can enter either the percentages or amounts, depending on the value you entered for your batch source. If you enter an amount, AutoInvoice requires that the distribution amounts sum to the amount of the transaction. If you enter a percent, AutoInvoice requires that the distribution percentages sum to 100 for each account class you pass.

Distributions in this table are linked to the appropriate transaction lines in the RA_INTERFACE_LINES_ALL table via the transaction flexfield. Though the distribution for 'REC' account class is at the invoice level, it may be linked to any transaction line of the invoice in RA_INTERFACE_LINES_ALL. AutoInvoice will then correctly transfer all distributions to RA_CUST_TRX_LINE_GL_DIST.

Column Name	Value
ACCOUNT_CLASS	<p>Enter the account class for this accounting distribution. AutoInvoice uses the account class you enter here to determine the type of account you are supplying for this accounting distribution. You must enter a value for this column.</p> <p>Validation: Must be either 'REV', 'FREIGHT', 'TAX', 'REC', 'CHARGES', 'UNBILL', or 'UNEARN'. If the transaction uses the 'Advance Invoice' invoicing rule, do not enter 'UNBILL' in this column. If the transaction uses the 'Arrears Invoice' invoicing rule, do not enter 'UNEARN' in this column</p> <p>Destination: RA_CUST_TRX_LINE_GL_DIST. ACCOUNT_CLASS</p>
ACCTD_AMOUNT	<p>This column is optional. If you enter 'AMOUNT' for your batch source option 'Revenue Account Allocation', then AutoInvoice will accept whatever is passed in this column without validation. If this column is null, then AutoInvoice will compute the accounted amount for this distribution line.</p> <p>Validation: None</p> <p>Destination: None</p>
AMOUNT	<p>Enter the amount for this accounting distribution.</p> <p>If this accounting distribution is for a transaction which does not use an accounting rule and depending on the value you entered for your batch source, you must enter either a value in this column or in PERCENT. If you specify the percent in your batch source, AutoInvoice computes the value in this column.</p> <p>Do not enter a value in this column if this accounting distribution is for a transaction which uses an accounting rule or if this distribution is a receivables ('REC') account. If this distribution is for a receivables account, you must enter 100 in RA_INTERFACE_DISTRIBUTIONS_ALL.PERCENT.</p> <p>Validation: If this transaction does not use an accounting rule, the sum of all distribution amounts for this transaction of a given line type must equal the amount for the transaction. AutoInvoice corrects amounts that have incorrect currency precision.</p>

	Destination:	RA_CUST_TRX_LINE_GL_DIST.AMOUNT
ATTRIBUTE1-15	Enter the Descriptive Flexfield attribute information for this accounting distribution. Descriptive Flexfield attributes allow you to store additional columns, the contents of which you define. These columns are optional.	
	Validation:	None
	Destination:	RA_CUST_TRX_LINE_GL_DIST.ATTRIBUTE1-15
ATTRIBUTE_ CATEGORY	Enter the Descriptive Flexfield category information for this accounting distribution. Descriptive Flexfield categories allow you to store different categories of attributes. This column is optional.	
	Validation:	None
	Destination:	RA_CUST_TRX_LINE_GL_DIST. ATTRIBUTE_CATEGORY
CODE_ COMBINATION_ID	Enter the code combination id of the Accounting Flexfield for this accounting distribution.	
	This column is optional. Depending on the value you entered for your batch source you must enter either a value in this column or a combination of segment values in SEGMENT1-30. If you specify the combination of segment values in your batch source, AutoInvoice defaults a value in this column.	
	Validation:	Must exist in GL_CODE_COMBINATIONS. CODE_COMBINATION_ID
	Destination:	RA_CUST_TRX_LINE_GL_DIST. CODE_COMBINATION_ID
COMMENTS	Enter comments about this accounting distribution. This column is optional.	
	Validation:	None
	Destination:	RA_CUST_TRX_LINE_GL_DIST.COMMENTS
INTERFACE_ DISTRIBUTION_ID	This column is used by AutoInvoice and should be left null. AutoInvoice defaults a value into this column using the sequence RA_CUST_TRX_LINE_GL_DIST_S. This is the primary key for RA_INTERFACE_DISTRIBUTIONS_ALL.	
	Validation:	None

	Destination: RA_CUST_TRX_LINE_GL_DIST. CUST_TRX_LINE_GL_DIST_ID
INTERFACE_LINE_ATTRIBUTE1-15	<p>Enter the same Line Transaction Flexfield for the transaction with which you want to associate this accounting distribution. You must enter a value for each attribute you enabled for the Line Transaction Flexfield.</p> <p>Validation: None</p> <p>Destination: None</p>
INTERFACE_LINE_CONTEXT	<p>This is a required column in AutoInvoice. Enter the context of the Line Transaction Flexfield entered in columns INTERFACE_LINE_ATTRIBUTE1-15.</p> <p>Validation: If you pass lines with global context set this column to 'Global Data Elements'</p> <p>Destination: RA_CUSTOMER_TRX_LINES>INTERFACE_LINE_CONTEXT</p>
INTERFACE_LINE_ID	<p>This column is used by AutoInvoice and should be left null. AutoInvoice defaults a value into this column using INTERFACE_LINE_ATTRIBUTE1-15 and INTERFACE_LINE_CONTEXT.</p> <p>Validation: None</p> <p>Destination: RA_CUST_TRX_LINE_GL_DIST. CUSTOMER_TRX_LINE_ID</p>
INTERFACE_STATUS	<p>This column is used by AutoInvoice and should be left null.</p> <p>Validation: None</p> <p>Destination: None</p>
LAST_UPDATE_LOGIN	<p>This column is used by AutoInvoice and should be left null. AutoInvoice updates this column when it selects rows from the RA_INTERFACE_DISTRIBUTIONS_ALL table for processing.</p> <p>Validation: None</p> <p>Destination: None</p>
ORG_ID	<p>If you use the Oracle Applications multiple organizations feature, enter the operating unit's org_id. If you do not use the Oracle Applications multiple organizations feature, leave this column null. To insert rows into open interface tables, use the multiple organization table RA_INTERFACE_DISTRIBUTIONS_ALL and explicitly populate the</p>

org_id column. You can determine an operating unit's org_id by querying the HR_OPERATING_UNITS view:

```
SELECT name, organization_id from  
HR_OPERATING_UNITS;
```

When you import rows *from* open interface tables, only rows from your current operating unit are processed.

PERCENT

Enter the percent for this accounting distribution.

If this accounting distribution is for a transaction which does not use an accounting rule and depending on the value you entered for your batch source, you must enter either a value in this column or in AMOUNT. If you specify the amount in your batch source, AutoInvoice defaults a value in this column.

If this accounting distribution is for a transaction which uses an accounting rule, you must enter a value in this column.

Validation: The sum of all accounting distribution percentages for a transaction must sum to 100 for an account class

Destination: RA_CUST_TRX_LINE_GL_DIST.PERCENT

REQUEST_ID

This column is used by AutoInvoice and should be left null.

Validation: None

Destination: None

SEGMENT1-30

Enter an Accounting Flexfield value to each segment you enable in your Oracle Receivables application. For example, if you enable six Accounting Flexfield segments, you must enter six values in columns SEGMENT1-6. Be sure to enter the correct segment value. For example, value '01' is not the same as '1'.

Depending on the value you entered for your batch source you must enter either a combination of segment values in these columns or a value in CODE_COMBINATION_ID.

Validation: Valid combination of Accounting Flexfield segment values must exist in GL_CODE_COMBINATIONS

Destination: None

RA_INTERFACE_ERRORS_ALL

This table is used to store information about interface data that failed validation. The information in this table will be used to generate the AutoInvoice Validation report. AutoInvoice identifies all errors for each transaction line, thus reducing multiple validation and correction cycles.

Column Name	Value
INTERFACE_ LINE_ID	<p>If both INTERFACE_SALESCREDIT_ID and INTERFACE_DISTRIBUTION_ID are null, then the row in RA_INTERFACE_LINES_ALL associated to this INTERFACE_LINE_ID failed validation.</p> <p>Validation: None</p> <p>Destination: None</p>
INTERFACE_ SALESCREDIT_ID	<p>If this column is not null, then the row in RA_INTERFACE_SALESCREDITS_ALL associated to this INTERFACE_SALESCREDIT_ID failed validation.</p> <p>Validation: None</p> <p>Destination: None</p>
INTERFACE_ DISTRIBUTION_ID	<p>If this column is not null, then the row in RA_INTERFACE_DISTRIBUTIONS_ALL associated to this INTERFACE_DISTRIBUTION_ID failed validation.</p> <p>Validation: None</p> <p>Destination: None</p>
INVALID_VALUE	<p>The invalid value that failed validation displays in this column, if applicable.</p> <p>Validation: None</p> <p>Destination: None</p>
LINK_TO_LINE_ID	<p>This column displays the INTERFACE_LINE_ID of the line to which this line that failed validation is linked. For example, you have a tax line that fails and is linked to an invoice line that fails, then this column stores the INTERFACE_LINE_ID of the invoice line.</p> <p>Validation: None</p> <p>Destination: None</p>

MESSAGE_TEXT

The message text is stored in this column.

Validation: None

Destination: None

ORG_ID

If you use the Oracle Applications multiple organizations feature, enter the operating unit's org_id. If you do not use the Oracle Applications multiple organizations feature, leave this column null. To insert rows into open interface tables, use the multiple organization table RA_INTERFACE_ERRORS_ALL and explicitly populate the org_id column. You can determine an operating unit's org_id by querying the HR_OPERATING_UNITS view:

```
SELECT name, organization_id from  
HR_OPERATING_UNITS;
```

When you import rows *from* open interface tables, only rows from your current operating unit are processed.

Processing Bank Receipts Using Oracle Receivables Applications AutoLockbox

Oracle Receivables and Oracle Government Receivables' AutoLockbox eliminates manual data entry by allowing you to automatically process receipts that are sent directly to your bank. You can determine how you want to apply these receipts to your customers' outstanding items.

Basic Needs

Your Oracle Receivables application provides you with the features you need to satisfy the following basic integration needs. You should be able to:

- Automatically record foreign and functional currency receipts from your bank.
- Automatically identify the customer who remitted the receipt.
- Update receipts that are rejected during the validation process.
- Interface any data that your bank can transmit.

Major Features

Historical Data Conversion

AutoLockbox can interface receipts from your previous accounting system, making the transition from your old accounting application to your Oracle Receivables application smooth and efficient.

Complete Validation

Your Oracle Receivables application ensures that the receipts you are interfacing are accurate and valid within your Oracle Receivables application.

AutoLockbox Processing Report

Your Oracle Receivables application prints a report for you to review after submitting AutoLockbox. You need to correct and resubmit records that have failed the import or validation step.

Flexible Transmission Formats

Transmission formats are used by your banks to determine what information to pass and how to format, your bank file. Your transmission format must match each bank control file that you create. Your Oracle Receivables application provides you with two sample formats, 'CONVERT, (arconv.ctl)' and 'DEFAULT, (ardeft.ctl)'. You can modify these transmission formats or create new ones.

Magnetic Ink Character Recognition (MICR) Numbers

Your Oracle Receivables application stores information about your customers and their associated MICR numbers. MICR numbers consist of the bank's transit routing number and the customer's bank account. MICR numbers are used to associate a customer with a receipt only if the customer's number is not included in the transmission format and if the MICR number is passed. If the MICR number is not passed, then the receipt is imported as Unidentified if AutoAssociate is not invoked. You can use the QuickCash, Enter Receipts or Reapply Receipts forms to assign customers to unidentified receipts. When a receipt is manually or automatically entered with a new MICR number, this number is stored and used for future reference.

AutoAssociate

Besides using customer and MICR numbers to identify receipts, you can use the invoices to which you apply receipts. If you include invoice numbers from your customers' remittance advice in the bank file, your Oracle Receivables application uses the invoice number for each receipt to identify the customer. This is called AutoAssociate. To invoke AutoAssociate you must set AutoAssociate to Yes in the Define Lockboxes form, ensure that invoice numbers are unique and all invoices applied to the receipt belong to one customer. If AutoLockbox is unable to identify the customer, the receipt remains with a status of Unidentified.

Associate Receipts With Billing Locations

Your Oracle Receivables application lets you track receipt activity for each of your customers' billing sites. If you want to use this feature you must include billing location in your transmission format and ensure that the Require Billing Location for Receipt in the Define System Options form is set to Yes. You can also require billing locations for specific lockboxes if you want to prevent unidentified receipts from being processed through AutoLockbox. The value that

you enter for any lockbox overrides the value at the system level. If the billing location is required and you do not import this information, AutoLockbox does not transfer the receipt. In this case, you must use the Maintain Lockbox Transmission Data form to enter the billing location for the receipt.

Your Oracle Receivables application allows you to import receipts without location even if the system option Require Billing Location for Receipt is set to Yes. You are required to enter the location information in the Quickcash form otherwise Post QuickCash will not process these receipts.

Overapplying Invoices

AutoLockbox does not validate the sign attributes that you assigned to your transaction type. If your application amount exceeds the balance due on your invoice, the invoice will be closed and the additional amount will remain on your receipt. If the sign of your application is different from the sign of the balance due on your invoice Lockbox will not make the application and the amount will again remain unapplied on the receipt.

Assigning Document Numbers to Receipts

Your Oracle Receivables application provides you with a way to automatically number receipts processed through AutoLockbox. Document numbers provide you with a detailed audit trail as well as inform you of any missing documents. To use this feature you must invoke the Sequential Numbering profile option and define and assign document sequences and categories to your payment methods.

Automatic Receipts

If you are using the automatic receipts feature, AutoLockbox will ignore all receivables items that are selected for Automatic receipt.

Receipts in Multiple Currencies

Your Oracle Receivables application lets you process receipts in multiple currencies. If you pass the currency code, exchange rate type and receipt date, AutoLockbox will try to determine the exchange rate. If it is unable to determine the exchange rate the receipt will fail validation.

Cross Currency Deposits

Your Oracle Receivables application now supports cross currency deposits. This implies that receipts in your lockbox can be either in the same currency as that of the bank account or in any other currency provided the bank account is in your functional currency and it's Receipts Multi-Currency field is set to Yes.

Audit Trails

Your Oracle Receivables application keeps track of each transmission that you submit. You can trace a receipt back to its transmission, lockbox and origination date.

Bank File Validation

Your Oracle Receivables application validates the data you receive from the bank to ensure that the entire file was received, that a receipt is not a duplicate within a batch, and that the customers and invoices are valid.

Definitions

AutoAssociate

An option that allows you to specify whether you want your Oracle Receivables application to determine the customer using invoice numbers if the customer cannot be identified from either the MICR number or the customer number. You can only use this feature if your bank transmits invoice numbers and these numbers are unique. Otherwise, your Oracle Receivables application treats the receipt as unidentified.

BAI

An acronym for the Banking Administration Institute. This organization has recommended a widely accepted format for sending lockbox data.

Bank File

The bank file is the data file you receive from the bank which contains all of the receipt information that the bank has deposited in your bank account.

Control File

The control file is used by SQL*Loader. It maps the data in your bank file to tables and columns in the ORACLE database. You must create a control file for each bank file that uses a different transmission format.

MICR Number

A MICR (Magnetic Ink Character Recognition) number relates your customer to a bank. This number consists of two segments. The first segment is the Transit Routing number, which identifies the bank from which your customer draws their check. The second segment identifies your customer's account at that bank.

Overflow Record

An Overflow record is a type of bank file record that stores additional receipt information that could not fit on the receipt record. Each Overflow record must have a receipt record as a parent. Typically, an Overflow record will store additional invoice numbers and the amount of the receipt to apply to each invoice.

Record Type

A bank file is made up of many different rows or records. Each record must have a type. For example, a record may store information about a receipt record or a batch record. Record types help your Oracle Receivables application determine where different types of data are stored in your bank file, and are specified in your transmission format.

Record Identifier

A record identifier consists of at most two characters which your Oracle Receivables application uses to identify each record type. For example, your Oracle Receivables application can identify a receipt record in BAI bank files because this record always starts with the character '6'.

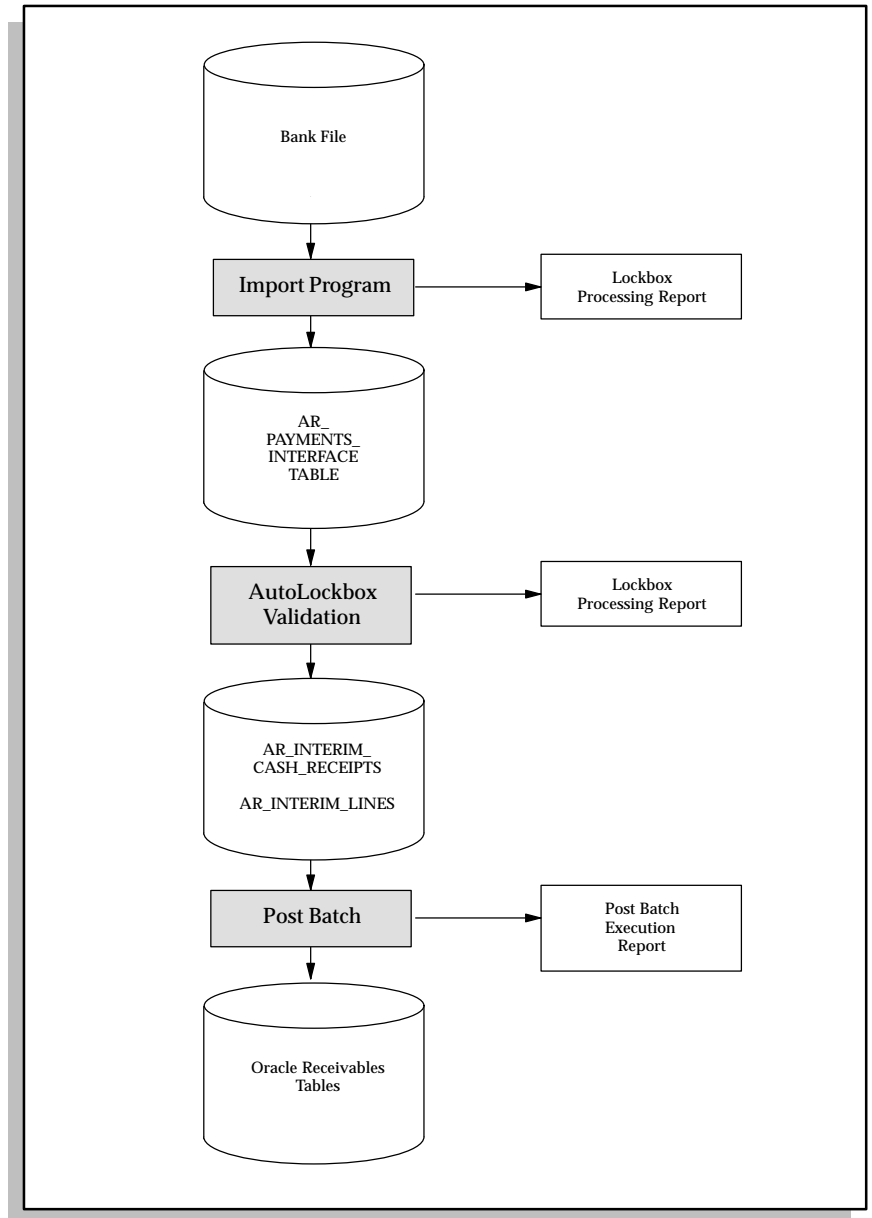
Transmission Format

A transmission format defines what data your bank is sending in the bank file, and how that data is organized. You must work with your bank to determine the content of your transmission format.

Overview

The following diagram shows how receipt data from your bank file is imported into your Oracle Receivables application tables.

Figure 10 – 5 Importing Data from your Bank File



Preparing Oracle Receivables and Oracle Government Receivables

You need to prepare your Oracle Receivables application for any new data that you want AutoLockbox to interface.

- Define AutoCash Rule Sets
- Define System Options
- Define Remittance Banks
- Define Receipt Classes
- Define Payment Methods
- Define Receipt Source
- Define Profile Options
- Define Lockboxes
- Define Transmission Formats

Bank File

When you receive your bank file be sure to name the file and move it to the appropriate directory. You will need to specify the location of your bank file when you submit AutoLockbox. If you receive daily files from your bank, take precaution not to overwrite the files from the previous day.

SQL*Loader Control File

Your Oracle Receivables application uses SQL*Loader to load information from your operating system files into your Oracle Receivables application database. For SQL*Loader to load your bank file properly, each logical record that your bank sends to you must end with a carriage return. If not, SQL*Loader displays an error message when you initiate AutoLockbox.

Additional Information: Part II SQL*Loader
(*ORACLE Utilities User's Guide Version 6.0*)

Importing Data From Your Bank File

AutoLockbox is a three step process. The first step involves reading and formatting data from your bank file into AutoLockbox tables. Your Oracle Receivables application provides you with two SQL*Loader

control files which you can use to execute this step. If you do not want to use these control files you can create your own. The second step involves submitting the validation program which checks data in the AutoLockbox tables for compatibility with your Oracle Receivables application. Once data is validated, it is transferred into QuickCash tables. The final step involves submitting the Post QuickCash process which updates your customer's balances. These steps can be submitted individually or together from the submit Lockbox Processing form.

AutoLockbox Validation

AutoLockbox validates all of your data for compatibility with your Oracle Receivables application. AutoLockbox validates your data by ensuring that the columns in AR_PAYMENTS_INTERFACE_ALL reference the appropriate values and columns in your Oracle Receivables application.

Transmission Level Validation

AutoLockbox validates your lockbox transmission to ensure that transmission information corresponds to your transmission format. The following attributes are validated to ensure that your transmission contains appropriate information for your Oracle Receivables application:

- Transmission format contains receipt records
- Lockbox number is part of the transmission format or you specify it when you submit AutoLockbox from the Run Lockbox Interface form
- GL date is in an open accounting period
- Total transmission record count and amount that you supply must match the actual receipt count and amount that is determined by AutoLockbox
- Origination number is valid if it is provided

Lockbox Level Validation

AutoLockbox validates your lockbox records to ensure that lockbox information corresponds to your transmission format. The following attributes are validated to ensure that your lockbox records contain appropriate information for your Oracle Receivables application:

- Lockbox number is specified in either the Lockbox Header or the Lockbox Trailer, and is valid
- Lockbox batch count is correct if it is provided
- Lockbox amount is correct if it is provided
- Lockbox record count is correct if it is provided
- Origination number is valid if it is provided
- No duplicate lockbox numbers

Batch Level Validation

AutoLockbox validates your batch records to ensure that batch information corresponds to your transmission format. The following attributes are validated to ensure that your lockbox records contain appropriate information for your Oracle Receivables application:

- Batch name exists on batch records
- Batch name is unique within the transmission
- Batch amount is correct
- Batch record count is correct
- Lockbox number exists on batch records if this number is part of the transmission format

Receipt Level Validation

AutoLockbox validates your receipt records to ensure that receipt information corresponds to your transmission format. The following attributes are validated to ensure that your receipt records contain appropriate information for your Oracle Receivables application:

- Remittance amount is specified
- Check number is specified
- Item number is specified and is unique within a batch, a lockbox or the transmission, depending on the transmission format
- Lockbox number is specified if this number is not part of the Lockbox Header or the Lockbox Trailer of the transmission format, and batches are not imported
- Batch name is specified if either Batch Headers or Batch Trailers are part of the transmission format

- Account number is specified if Transit Routing Number is part of the transmission format
- Invoice1-8 are either valid invoice numbers or are left blank
- Installment1-8 are either valid installment numbers or are left blank.
- Invoice, debit memo, credit memo, deposit, on-account credit, or chargeback number does not belong to a guarantee or receipt
- Invoice number is entered where an application amount is specified.
- Sum of all of the Amount Applied columns for a receipt does not exceed the remittance amount
- Customer number is valid
- Customer number and MICR number both reference the same customer, if both are provided
- Receipt date is specified
- Payment method is valid
- Currency is valid

Overflow Level Validation

AutoLockbox validates your overflow records to ensure that overflow information corresponds to your transmission format. The following attributes are validated to ensure that your overflow records contain appropriate information for your Oracle Receivables application:

- Batch name is specified if either Batch Headers or Batch Trailers are part of the transmission format
- Lockbox number is specified if either the Batch Header or the Batch Trailer are not specified and the transmission format includes lockbox number
- Item number is specified and matches a receipt record
- Overflow indicator is specified, unless it is the last overflow record
- Overflow sequence is specified
- Invoice1-8 are valid invoice numbers. These numbers are optional, and can be left blank.

- Installment1-8 are either valid installment numbers or are left blank.
- Invoice number is entered where an application amount is specified.

Customer Validation

AutoLockbox has a number of ways to determine the customer for the receipt records you import. Depending upon your transmission format and how you prepare your Oracle Receivables application, AutoLockbox can either validate your customer data based on the following attributes, or interface the receipt as unidentified if no match is found:

- Customer number is valid
- MICR number is valid
- Bill-To customer is from an AutoAssociated invoice (if AutoAssociate is enabled)

Using Lockbox

Running Lockbox

You submit the import step of AutoLockbox using the Run Lockbox Interface form. You may choose to import, validate and run PostQuickCash all in one step or perform the steps separately using the same form.

Reviewing the Lockbox Processing Report After Import

Each time you submit the import step of AutoLockbox, your Oracle Receivables application automatically generates the submit import section of the Lockbox Processing report. This report displays the number of records that were successfully imported. If you use SQL*Loader as your import program it always creates a .log file which can be found in the \$AR_TOP/out directory. The .log file contains general information about the activity of your SQL* Loader run. SQL*Loader can also create a .dis and .bad file in the same directory, if it has records to write to these files. The .bad file contains information about any records that were rejected due to Oracle errors such as an invalid date. The .dis file contains discarded records that did not satisfy any of the WHEN clauses in your control file. Your Oracle

Receivables application prints a line on the bottom of the Lockbox Processing report informing you of any rejected or discarded files. The actual file will contain the record rejected or discarded. The log file will contain additional information including the reason for the record not being imported.

Reviewing the Lockbox Processing Report After Validation

Each time you submit the validation step of AutoLockbox processing, your Oracle Receivables application automatically generates the Submit Validation section of the Lockbox Processing report with the Submit Validation section. This report displays the number of records that have either passed or failed validation. For records that passed validation your Oracle Receivables application creates QuickCash receipt batches.

Correcting Records that Failed Validation

Your Oracle Receivables application provides you with the Maintain Lockbox Transmission form to correct records that failed validation. Once these records are updated they can be resubmitted for validation.

Reviewing Validated Receipts

After you import and validate your receipts, you can review these receipts in the QuickCash form. If your transmission contains receipts that failed validation, you can still review all of the receipts that were successfully validated. You can use the Transmission Request field in the QuickCash form to query all receipt batches that are included in one transmission. Your Oracle Receivables application lets you update or delete any receipt information, just as if you manually entered the receipt.



QuickCash

(Oracle Receivables Reference Manual or Oracle Government Receivables Reference Manual)

Updating Your Receivable Balance

Once receipts have been validated, you must submit the Post QuickCash program to update your customers receivable balances. You can submit this program through the Run Lockbox Interface or through the Post QuickCash forms.

Oracle Receivables and Oracle Government Receivables Tables

Lockbox transfers receipt data into the following Oracle Receivables applications Quickcash tables:

AR_INTERIM_CASH_RECEIPTS

AR_INTERIM_LINES

When you run PostQuickCash, the receipt data is transferred from the QuickCash tables to the following Oracle Receivables applications tables:

AR_CASH_RECEIPTS

AR_RECEIVABLES_APPLICATIONS

AR_CASH_RECEIPT_HISTORY

Table and Column Descriptions

Below is a detailed description of the AR_PAYMENTS_INTERFACE_ALL table your Oracle Receivables application uses to store receipt data from you bank file. Each column has important detail information you need to know to successfully run Lockbox. The Destination column gives you the interim QuickCash tables and the actual Oracle Receivables applications tables to which the data is transferred to from AR_PAYMENTS_INTERFACE_ALL.

Understanding the AR_PAYMENTS_INTERFACE_ALL Table

The AR_PAYMENTS_INTERFACE_ALL table is where your Oracle Receivables application stores AutoLockbox receipt data that you import from your bank file. The AR_PAYMENTS_INTERFACE_ALL table contains the following columns:

Column Name		Null?	Type
TRANSMISSION_RECORD_ID			NUMBER(15)
SOURCE:	AR_PAYMENTS_INTERFACE_ALL.S.NEXTVAL		
DESTINATION:	None		
CREATION_DATE			DATE

Table 10 – 9 AR_PAYMENTS_INTERFACE_ALL Table (Page 1 of 10)

Column Name		Null?	Type
SOURCE:	current system date		
DESTINATION:	None		
CREATED_BY			NUMBER(15)
SOURCE:	FND_USER.USER_ID		
DESTINATION:	AR_BATCHES.CREATED_BY		
	AR_INTERIM_CASH_RECEIPTS.CREATED_BY		
	AR_INTERIM_CASH_RECEIPT_LINES.CREATED_BY		
LAST_UPDATE_LOGIN			NUMBER(15)
SOURCE:	UNKNOWN		
DESTINATION:	None		
LAST_UPDATED_BY			NUMBER(15)
SOURCE:	FND_USER.USER_ID		
DESTINATION:	None		
LAST_UPDATE_DATE			DATE
SOURCE:	CURRENTSYSTEM DATE		
DESTINATION:	None		
RECORD_TYPE		NOT NULL	VARCHAR2(2)
SOURCE:	AR_TRANS_RECORD_FORMATS.RECORD_IDENTIFIER		
DESTINATION:	None		
STATUS			VARCHAR2(30)
SOURCE:	FND_MESSAGES.MESSAGE_NAME		
DESTINATION:	None		
TRANSMISSION_REQUEST_ID			NUMBER(15)
SOURCE:	FND_CONCURRENT_REQUESTS.REQUEST_ID		
DESTINATION:	None		
TRANSMISSION_ID			NUMBER(15)
SOURCE:	AR_TRANSMISSIONS.TRANSMISSION_ID		
DESTINATION:	None		

Table 10 – 9 AR_PAYMENTS_INTERFACE_ALL Table (Page 2 of 10)

Column Name		Null?	Type
DESTINATION_ACCOUNT			VARCHAR2(25)
SOURCE:	LOCKBOX DATA FILE OR ENTERED BY USER VIA		
	'MAINTAIN LOCKBOX TRANSMISSION DATA'		
DESTINATION:	AR_TRANSMISSIONS.DESTINATION		
ORIGINATION			VARCHAR2(25)
SOURCE:	LOCKBOX DATA FILE OR ENTERED BY USER VIA		
	'MAINTAIN LOCKBOX TRANSMISSION DATA'		
DESTINATION:	AR_TRANSMISSIONS.ORIGIN		
DEPOSIT_DATE			DATE
SOURCE:	LOCKBOX DATA FILE OR ENTERED BY USER VIA		
	'MAINTAIN LOCKBOX TRANSMISSION DATA'		
DESTINATION:	AR_BATCHES.DEPOSIT_DATE		
GL_DATE			DATE
SOURCE:	DERRIVED FROM DEPOSIT DATE, IMPORT DATE OR		
	ENTERED DATE		
DESTINATION:	AR_BATCHES.GL_DATE		
	AR_INTERIM_CASH_RECEIPTS.GL_DATE		
	AR_CASH_RECEIPT_HISTORY.GL_DATE		
DEPOSIT_TIME			VARCHAR2(8)
SOURCE:	LOCKBOX DATA FILE OR ENTERED BY USER VIA		
	'MAINTAIN LOCKBOX TRANSMISSION DATA'		
DESTINATION:	None		
TRANSMISSION_RECORD_COUNT			NUMBER(15)
SOURCE:	LOCKBOX DATA FILE OR ENTERED BY USER VIA		
	'MAINTAIN LOCKBOX TRANSMISSION DATA'		
DESTINATION:	AR_TRANSMISSIONS.COUNT		
TRANSMISSION_AMOUNT			NUMBER

Table 10 – 9 AR_PAYMENTS_INTERFACE_ALL Table (Page 3 of 10)

Column Name		Null?	Type
SOURCE:	LOCKBOX DATA FILE OR ENTERED BY USER VIA		
	'MAINTAIN LOCKBOX TRANSMISSION DATA'		
DESTINATION:	AR_TRANSMISSIONS.AMOUNT		
TRANSFERRED_RECEIPT_COUNT			NUMBER
SOURCE:	PROGRAM COUNTS NUMBER OF RECORDS TRANSFERRED		
	SUCCESSFULLY		
DESTINATION:	AR_TRANSMISSIONS.VALIDATED_COUNT		
TRANSFERRED_RECEIPT_AMOUNT			NUMBER
SOURCE:	PROGRAM COUNTS RECEIPT AMOUNTS OF RECORDS		
	SUCCESSFULLY TRANSFERRED		
DESTINATION:	AR_TRANSMISSIONS.VALIDATED_AMOUNT		
LOCKBOX_NUMBER			VARCHAR2(30)
SOURCE:	PROVIDED BY BANK OR ENTERED BY USER AT RUNTIME		
DESTINATION:	None		
LOCKBOX_BATCH_COUNT			NUMBER
SOURCE:	LOCKBOX DATA FILE OR ENTERED BY USER VIA		
	'MAINTAIN LOCKBOX TRANSMISSION DATA'		
DESTINATION:	None		
LOCKBOX_RECORD_COUNT			NUMBER
SOURCE:	LOCKBOX DATA FILE OR ENTERED BY USER VIA		
	'MAINTAIN LOCKBOX TRANSMISSION DATA'		
DESTINATION:	None		
LOCKBOX_AMOUNT			NUMBER
SOURCE:	LOCKBOX DATA FILE OR ENTERED BY USER VIA		
	'MAINTAIN LOCKBOX TRANSMISSION DATA'		
DESTINATION:	None		

Table 10 – 9 AR_PAYMENTS_INTERFACE_ALL Table (Page 4 of 10)

Column Name		Null?	Type
BATCH_NAME			VARCHAR2(25)
SOURCE:	LOCKBOX DATA FILE OR ENTERED BY USER VIA		
	'MAINTAIN LOCKBOX TRANSMISSION DATA'		
DESTINATION:	AR_BATCHES.LOCKBOX_BATCH_NAME		
BATCH_AMOUNT			NUMBER
SOURCE:	LOCKBOX DATA FILE OR ENTERED BY USER VIA		
	'MAINTAIN LOCKBOX TRANSMISSION DATA'		
DESTINATION:	AR_BATCHES.CONTROL_AMOUNT		
BATCH_RECORD_COUNT			NUMBER(15)
SOURCE:	LOCKBOX DATA FILE OR ENTERED BY USER VIA		
	'MAINTAIN LOCKBOX TRANSMISSION DATA'		
DESTINATION:	AR_BATCHES.CONTROL_COUNT		
ITEM_NUMBER			NUMBER
SOURCE:	LOCKBOX DATA FILE OR ENTERED BY USER VIA		
	'MAINTAIN LOCKBOX TRANSMISSION DATA'		
DESTINATION:	None		
CURRENCY_CODE			VARCHAR2(15)
SOURCE:	LOCKBOX DATA FILE OR ENTERED BY USER VIA		
	'MAINTAIN LOCKBOX TRANSMISSION DATA'		
DESTINATION:	AR_BATCHES.CURRENCY_CODE		
	AR_INTERIM_CASH_RECEIPTS.CURRENCY_CODE		
EXCHANGE_RATE			NUMBER
SOURCE:	LOCKBOX DATA FILE OR ENTERED BY USER VIA		
	'MAINTAIN LOCKBOX TRANSMISSION DATA'		
DESTINATION:	AR_BATCHES.EXCHANGE_RATE		
	AR_INTERIM_CASH_RECEIPTS.EXCHANGE_RATE		
EXCHANGE_RATE_TYPE			VARCHAR2(30)

Table 10 – 9 AR_PAYMENTS_INTERFACE_ALL Table (Page 5 of 10)

Column Name		Null?	Type
SOURCE:	DEFAULTS FROM LOCKBOX DEFINITIONS OR ENTERED BY		
	USER VIA 'MAINTAIN LOCKBOX TRANSMISSION DATA'		
DESTINATION	AR_BATCHES.EXCHANGE_RATE_TYPE		
	AR_INTERIM_CASH_RECEIPTS.EXCHANGE		
	AR_RATE_TYPE		
REMITTANCE_AMOUNT			NUMBER
SOURCE:	LOCKBOX DATA FILE OR ENTERED BY USER VIA		
	'MAINTAIN LOCKBOX TRANSMISSION DATA'		
DESTINATION:	AR_INTERIM_CASH_RECEIPTS.AMOUNT		
TRANSIT_ROUTING_NUMBER			VARCHAR2(25)
SOURCE:	LOCKBOX DATA FILE OR ENTERED BY USER VIA		
	'MAINTAIN LOCKBOX TRANSMISSION DATA'		
DESTINATION:	AP_BANK_BRANCHES.BANK_NAME		
	AP_BANK_BRANCHES.BANK_BRANCH_NAME		
	AP_BANK_BRANCHES.BANK_NUM		
ACCOUNT			VARCHAR2(30)
SOURCE:	LOCKBOX DATA FILE OR ENTERED BY USER VIA		
	'MAINTAIN LOCKBOX TRANSMISSION DATA'		
DESTINATION:	AP_BANK_ACCOUNTS.BANK_ACCOUNT_NUM		
CUSTOMER_BANK_AC-COUNT_ID			NUMBER(15)
SOURCE:	AP_BANK_ACCOUNT_USES.EXTERNAL_BANK_		
	ACCOUNT_ID		
DESTINATION:	AR_INTERIM_CASH_RECEIPTS.CUSTOMER_BANK_		
	ACCOUNT_ID		
	AR_CASH_RECEIPTS.CUSTOMER_BANK_ACCOUNT_ID		

Table 10 – 9 AR_PAYMENTS_INTERFACE_ALL Table (Page 6 of 10)

Column Name		Null?	Type
CHECK_NUMBER			VARCHAR2(30)
SOURCE:	LOCKBOX DATA FILE OR ENTERED BY USER VIA		
	'MAINTAIN LOCKBOX TRANSMISSION DATA'		
DESTINATION:	AR_INTERIM_CASH_RECEIPTS.RECEIPT_NUMBER		
	AR_CASH_RECEIPTS.RECEIPT_NUMBER		
SPECIAL_TYPE			VARCHAR2(20)
SOURCE:	PROGRAM DETERMINES THE TYPE		
DESTINATION:	AR_INTERIM_CASH_RECEIPTS.SPECIAL_TYPE		
CUSTOMER_NUMBER			VARCHAR2(30)
SOURCE:	LOCKBOX DATA FILE OR ENTERED BY USER VIA		
	'MAINTAIN LOCKBOX TRANSMISSION DATA'		
DESTINATION:	NONE		
CUSTOMER_ID			NUMBER(15)
SOURCE:	PROGRAM DETERMINES IT		
DESTINATION:	AR_INTERIM_CASH_RECEIPTS.PAY_FROM_CUSTOMER		
	AR_CASH_RECEIPTS.PAY_FROM_CUSTOMER		
BILL_TO_LOCATION			VARCHAR2(40)
SOURCE:	LOCKBOX DATA FILE OR ENTERED BY USER VIA		
	'MAINTAIN LOCKBOX TRANSMISSION DATA'		
DESTINATION:	NONE		
CUSTOMER_SITE_USE_ID			NUMBER(15)
SOURCE:	PROGRAM DETERMINES IT		
DESTINATION:	AR_INTERIM_CASH_RECEIPTS.SITE_USE_ID		
	AR_CASH_RECEIPTS.CUSTOMER_SITE_USE_ID		
RECEIPT_DATE			DATE
SOURCE:	LOCKBOX DATA FILE OR ENTERED BY USER VIA		

Table 10 – 9 AR_PAYMENTS_INTERFACE_ALL Table (Page 7 of 10)

Column Name	Null?	Type
		'MAINTAIN LOCKBOX TRANSMISSION DATA'
DESTINATION:	AR_INTERIM_CASH_RECEIPTS.RECEIPT_DATE	
	AR_INTERIM_CASH_RECEIPTS.EXCHANGE_DATE	
	AR_CASH_RECEIPTS.RECEIPT_DATE	
	AR_CASH_RECEIPTS.EXCHANGE_DATE	
RECEIPT_METHOD		VARCHAR2(30)
SOURCE:	LOCKBOX DATA FILE OR ENTERED BY USER VIA	
	'MAINTAIN LOCKBOX TRANSMISSION DATA'	
DESTINATION:	NONE	
RECEIPT_METHOD_ID		NUMBER(15)
SOURCE:	PROGRAM DETERMINES IT	
DESTINATION:	AR_INTERIM_CASH_RECEIPTS.RECEIPT_METHOD_ID	
	AR_CASH_RECEIPTS.RECEIPT_METHOD_ID	
OVERFLOW_INDICATOR		VARCHAR2(1)
SOURCE:	LOCKBOX DATA FILE OR ENTERED BY USER VIA	
	'MAINTAIN LOCKBOX TRANSMISSION DATA'	
DESTINATION:	NONE	
OVERFLOW_SEQUENCE		NUMBER
SOURCE:	LOCKBOX DATA FILE OR ENTERED BY USER VIA	
	'MAINTAIN LOCKBOX TRANSMISSION DATA'	
DESTINATION:	NONE	
INVOICE1-8		VARCHAR2(20)
SOURCE:	LOCKBOX DATA FILE OR ENTERED BY USER VIA	
	'MAINTAIN LOCKBOX TRANSMISSION DATA'	
DESTINATION:	NONE	
AMOUNT_APPLIED1-8		NUMBER

Table 10 – 9 AR_PAYMENTS_INTERFACE_ALL Table (Page 8 of 10)

Column Name		Null?	Type
SOURCE:	LOCKBOX DATA FILE OR ENTERED BY USER VIA		
	'MAINTAIN LOCKBOX TRANSMISSION DATA'		
DESTINATION:	AR_INTERIM_CASH_RECEIPT_LINES.PAYMENT		
	_AMOUNT		
	AR_RECEIVABLE_APPLICATIONS.AMOUNT_APPLIED		
INVOICE1_STATUS-IN-VOICE8_STATUS			VARCHAR2(30)
SOURCE:	PROGRAM DETERMINES IT		
DESTINATION:	NONE		
COMMENTS			VAR-CHAR2(240)
SOURCE:	ENTERED BY USER VIA		
	'MAINTAIN LOCKBOX TRANSMISSION DATA'		
DESTINATION:	AR_BATCHES.COMMENTS		
	AR_INTERIM_CASH_RECEIPTS.COMMENTS		
ATTRIBUTE_CATEGORY			VARCHAR2(30)
ATTRIBUTE1-15			CHAR(40)
SOURCE:	LOCKBOX DATA FILE OR ENTERED BY USER VIA		
	'MAINTAIN LOCKBOX TRANSMISSION DATA'		
DESTINATION:	AR_INTERIM_CASH_RECEIPTS.ATTRIBUTE1...15		
	AR_CASH_RECEIPTS.ATTRIBUTE1...15		
INVOICE1_INSTALLMENT-			
INVOICE8_INSTALLMENT			NUMBER
SOURCE:	LOCKBOX DATA FILE OR ENTERED BY USER VIA		
	'MAINTAIN LOCKBOX TRANSMISSION DATA'		
DESTINATION:	NONE		
ORG_ID			NUMBER

Table 10 – 9 AR_PAYMENTS_INTERFACE_ALL Table (Page 9 of 10)

Column Name	Null?	Type
SOURCE:	ENTERED BY USER	
DESTINATION:	NONE	

Table 10 – 9 AR_PAYMENTS_INTERFACE_ALL Table (Page 10 of 10)

Assigning Values to Columns

You must assign values to all of the following columns in the AR_PAYMENTS_INTERFACE_ALL table in order for AutoLockbox to successfully convert data into receipts.

<u>Column Name</u>	<u>Value</u>
STATUS	Enter the value AR_PLB_NEW_RECORD for all records inserted into this table. The sample SQL*Loader control files provided by your Oracle Receivables application fill this column in for you.
DEPOSIT_DATE	Enter the date on which this transmission was actually deposited into your bank account. This date can be on any of the record types in your transmission. Each unique deposit date determines a batch of transmission records. For example, if you enter two unique deposit dates for your transmission, AutoLockbox divides your transmission into two batches of receipts.
RECORD_TYPE	Identify your record type. For example, if this is a batch header record, and your bank uses the value 3 to identify batch headers, enter 3 in this column. Find out from your bank what character they use to identify each one. Keep in mind that not all banks use all of the record types. Assign values to identify the following types of records: TRANSMISSION HEADER TRANSMISSION TRAILER LOCKBOX HEADERS LOCKBOX TRAILERS BATCH HEADERS BATCH TRAILERS PAYMENT RECORDS

PAYMENT OVERFLOW RECORDS SERVICE HEADER

ORG_ID

If you use the Oracle Applications multiple organizations feature, enter the operating unit's org_id. If you do not use the Oracle Applications multiple organizations feature, leave this column null. To insert rows into open interface tables, use the multiple organization table AR_PAYMENTS_INTERFACE_ALL and explicitly populate the org_id column. You can determine an operating unit's org_id by querying the HR_OPERATING_UNITS view:

```
SELECT name, organization_id from  
HR_OPERATING_UNITS;
```

When you import rows *from* open interface tables, only rows from your current operating unit are processed.

Your Oracle Receivables application lets you determine what information you want to include in you header, trailer and receipt records. You can reference any of the above types when you define the different records for your transmission format. Below we give you examples of how you might want to define these.

Assigning Values to Transmission Header and Trailer Records

If your record type is either a Transmission Header or a Transmission Trailer, then enter the following columns with the values you described. Transmission Headers and Trailers mark the beginning and ends of a specific data file. They usually contain information such as destination account, origination number, deposit date, and deposit time. You may have a Transmission Header without a Transmission Trailer, AutoLockbox does not require that you specify either of these record types in your transmission format. For each transmission you can only have one transmission header and one transmission trailer.

<u>Column Name</u>	<u>Value</u>
TRANSMISSION_RECORD_COUNT	Enter the number of records that you are importing. Include all of the types of records in the count; headers, trailers, receipts and overflow records.

TRANSMISSION_AMOUNT	Enter the amount of the transmission. This is the sum of all of the receipt amounts within the transmission.
DESTINATION_ACCOUNT	Enter your account number at the sending bank.
ORIGINATION	Enter the sending bank's transit routing number.
DEPOSIT_DATE	Enter the date this transmission was actually deposited in your bank account. When you use SQL*Loader to import your data, it converts the date to the ORACLE date format.
DEPOSIT_TIME	Enter the time the deposit was made.

Assigning Values to Lockbox Header or Trailer Records

If your record type is either a Lockbox Header or a Lockbox Trailer, enter the following columns with the values described. Lockbox Headers usually mark the beginning of a specific lockbox and contain information such as the destination account and origination number. Lockbox Trailers mark the end of specific lockboxes and contain information such as lockbox number, deposit date, lockbox amount and lockbox record count. Although you may have a Lockbox Header without a Lockbox Trailer, AutoLockbox does not require that you specify either of these record types in your transmission format.

<u>Column Name</u>	<u>Value</u>
LOCKBOX_NUMBER	Enter the lockbox name or number that your bank specifies. This is the same value that you entered in the Define Lockboxes form. LOCKBOX_NUMBER is mandatory on all Lockbox Headers and Trailers.
LOCKBOX_BATCH_COUNT	Enter the number of batches in this lockbox.
LOCKBOX_RECORD_COUNT	Enter the number of Payment records in this lockbox. Do not include Payment Overflow records.
LOCKBOX_AMOUNT	Enter the total value of the receipts in this lockbox.

DESTINATION_ACCOUNT	Enter your account number at the sending bank. If this value is included in a Transmission Header or Trailer, you must enter the same value.
ORIGINATION	Enter the sending bank's transit routing number. If this value is included in a Transmission Header or Trailer, you must have the same value here.

Assigning Values to Batch Header and Trailer Records

If your record type is either a Batch Header or a Batch Trailer, you can enter the following columns with the values described below. Batch Headers mark the beginning of a specific batch and contain information such as batch number deposit date and lockbox number. Batch Trailers mark the end of a specific batch and contain information such as batch number, lockbox number, batch record amount and batch amount. Although you may have a Batch Header without a Batch Trailer, AutoLockbox does not require that you specify either of these record types in your transmission format.

<u>Column Name</u>	<u>Value</u>
BATCH_NAME	Enter the name or number that the bank uses to identify the batch. This is required for each Batch Header and Trailer record. Oracle Receivables creates new batches for every
BATCH_AMOUNT	Enter the total value of all receipts in this batch.
BATCH_RECORD_COUNT	Enter the number of receipt records in this batch.
LOCKBOX_NUMBER	Enter the lockbox number assigned to receipts in this batch. If the lockbox number is included in your format, it must appear on every batch record.
COMMENTS	Enter any free-form comments about this batch.

Assigning Values to Receipt Records

If your record type is a Payment, you can enter the following columns with the values described below. A Payment record usually contains

information such as MICR number, batch number, item number, check number and remittance amount. Some of the values are mandatory for a Payment record, while others are optional. Every transmission must have Payment records.

<u>Column Name</u>	<u>Value</u>
LOCKBOX_NUMBER	Enter the lockbox number assigned to your receipts. If the lockbox number is included in your format and you do not have batch records, it must be entered for every receipt record.
BATCH_NAME	Enter the batch name for this receipt. If batch name is included in your format, it must be entered for every receipt record. Each unique batch name determines a batch of transmission records. For example, if you enter two unique batch names for your transmission, AutoLockbox divides your transmission into two batches of receipts.
ITEM_NUMBER	Enter a sequential number to indicate the location of this receipt in this batch. You must enter a value even if your format does not have batch, lockbox or transmission records. Item Number must be unique within a batch, a lockbox (if batches are not provided) or within a transmission (if neither batches nor lockboxes are provided).
REMITTANCE_AMOUNT	Enter the value of the receipt. You must enter a value for each receipt record.
CURRENCY_CODE	Enter the currency code for each receipt. Your Oracle Receivables application supports AutoLockbox Transmission receipts in different currencies.
EXCHANGE_RATE	Enter the exchange rate you want your Oracle Receivables application to use for this currency.
EXCHANGE_RATE_TYPE	Enter the type of exchange rate you are using for this receipt. You can enter Corporate, Spot or User.
RECEIPT_DATE	Enter the date that is written on your check.
RECEIPT_METHOD	Enter the payment method that you want to associate with this receipt. Payment methods contain information about your bank, bank account and receipt accounts. This payment method must be the same as the one you assigned to the batch source for this lockbox.

CHECK_ NUMBER	Enter the number printed on the receipt. You must enter a value for each receipt record.
TRANSIT_ ROUTING_ NUMBER	Enter the transit routing number from the receipt. This is optional but you must enter this number if you enter the account number. Your Oracle Receivables application uses transit routing number and account number together to identify the customer.
ACCOUNT	Enter the account number from the receipt. This is optional but you must enter this number if you enter the transit routing number.
CUSTOMER_ NUMBER	Enter the number assigned to your customer. This is optional.
INVOICE1-8	Enter the invoice numbers to which you apply this receipt. You do not have to start with INVOICE1, nor use all eight of the INVOICE columns on a record before you create a receipt record. You may find a list of valid values in AR_PAYMENT_SCHEDULES.TRX_NUMBER. Do not look at transactions whose class is (PMT or GUAR). Invoice numbers are optional.
INVOICE1-8_IN STALLMENT	Enter the installment number if your invoice has multiple payment schedules. If you do not specify the installment number for an invoice with multiple payment schedules, then your Oracle Receivables application will apply to the oldest payment schedule first. The installment number must be on the same record as the associated invoice number.
AMOUNT_ APPLIED1-8	Enter the amount of the receipt to apply to the invoice. You may specify invoice numbers without specifying the amount applied to each of these invoices. If you specify invoice numbers without specifying the amount applied to each invoice, your Oracle Receivables application applies the receipt to the invoices starting with the oldest receipt schedule first. The amount applied column's value must be on the same record as the invoice number to which it is applied. For example, you cannot have all of the invoice numbers on the receipt record and all of the

	amounts applied on the overflow. Applied amounts are optional.
COMMENTS	Enter any free-form comments about this receipt. Your Oracle Receivables application stores this data, but does not display these comments in any of the receipt entry forms.
ATTRIBUTE_CATEGORY	Enter the Descriptive Flexfield category information for this receipt.
ATTRIBUTE1-15	Enter the Descriptive Flexfield attributes for this category. You can use this column to transfer additional information about your receipt. For example, if your bank enters and transmits customer name, you can use an attribute column to import this name. The attributes are visible as Descriptive Flexfields in the Receipt zone of all the receipt entry forms.
BILL_TO_LOCATION	If you want to associate receipts with specific customer sites, enter the billing address for this receipt and include billing location in your transmission format. If the Site Required for Cash and Customer field in the Define System Options form is set to Yes, you must enter a value here. In addition, you can set the Require Billing Location field to Yes in the Define Lockbox form to require a billing location for a specific lockbox. The lockbox field will override the system level field.

Assigning Values To Overflow Records

If your record type is an Overflow record, enter the following columns with the values described. Some of these values are mandatory, while others are optional. Overflow records allow you to transmit additional information about a receipt that does not fit on the receipt record such as batch number, item number, sequence number, invoice number, debit memo number or debit item amounts. The most common use for this record type is to import additional invoice numbers to which the receipt should be applied.

<u>Column Name</u>	<u>Value</u>
LOCKBOX_NUMBER	Enter the number of the lockbox for this receipt. If the lockbox number is included in your format and you do not have any batch records, you must enter this number for each receipt and overflow record.

BATCH_NAME	Enter the batch for this overflow record. If the batch name is included in your format, you must enter this name for each overflow record.
ITEM_NUMBER	Enter a sequential number to indicate the location of the overflow record in this batch. All overflow records for a receipt have the same item number as the receipt record. You must enter an item number for each overflow record to reference the receipt.
OVERFLOW_INDICATOR	Your Oracle Receivables application uses this column to indicate overflow records for the current receipt. You determine your overflow indicator in encounters a change in the batch nameyour transmission format. To identify the last overflow record enter a value that is different from your overflow indicator. For example, in the BAI transmission format, '0' indicates an overflow record. You have three overflow records for a receipt, the first two records have '0' as the overflow indicator and the third record has '9'. Since the third record is not '0' it is identified as the last overflow record. You must enter a value for all overflow records.
OVERFLOW_SEQUENCE	Enter a sequential number to indicate the order of overflow records. Within each receipt, the Overflow Sequence usually begins with 1.
INVOICE1-8	Enter the invoice numbers to which you apply this receipt. You do not have to start with INVOICE1, nor use all eight of the INVOICE columns on a record before you create an overflow record. You may find a list of valid values in AR_PAYMENT_SCHEDULES.TRX_NUMBER. Do not look at transactions whose class is (PMT or GUAR). You may supply invoice numbers without specifying the amount applied to each invoice. Invoice numbers are optional.
INVOICE1-8_INSTALLMENT	Enter the installment number if your invoice has multiple payment schedules. If you do not specify the installment number for an invoice with multiple payment schedules, then your Oracle Receivables application will apply to the oldest payment schedule first. The installment number

must be on the same record as the associated invoice number.

**AMOUNT_
APPLIED1-8**

Enter the amount of the receipt to apply to the invoice. If you specify invoice numbers without specifying the amount applied to each invoice, your Oracle Receivables application applies the receipt to the invoices starting with the oldest receipt first. The amount applied column's value must be on the same record as the invoice number to which the receipt amount is applied.

System Assigned Columns

Your Oracle Receivables application assigns values to the columns listed below during the import processes. Your import file must leave these columns blank.

<u>Column Name</u>	<u>Null?</u>	<u>Type</u>
TRANSMISSION_RECORD_ID		NUMBER
CREATION_DATE		DATE
CREATED_BY		NUMBER
LAST_UPDATE_LOGIN		NUMBER
LAST_UPDATED_BY		NUMBER
LAST_UPDATE_DATE		DATE
TRANSMISSION_REQUEST_ID		NUMBER
CUSTOMER_ID		NUMBER
SPECIAL_TYPE		CHAR(20)
GL_DATE		DATE
STATUS		CHAR(30)
INVOICE1-8_STATUS		CHAR(30)
RECEIPT_METHOD_ID		NUMBER(15)
TRANSMISSION_ID		NUMBER(15)
INVOICE1-8_STATUS		VARCHAR2(30)
CUSTOMER_BANK_ACCOUNT_ID		NUMBER(15)
CUSTOMER_SITE_USE_ID		NUMBER(15)
TRANSFERRED_RECEIPT_COUNT		NUMBER
TRANSFERRED_RECEIPT_AMOUNT		NUMBER

Commonly Asked Questions

When applying a receipt to an invoice through Autolockbox will the Post QuickCash program automatically take the discount?

Autolockbox does not realize discounts. This is an operation of the Post QuickCash program

If the customer's Credit Profile and Payment terms are set to allow discounts, Post QuickCash will automatically take the discount. The discount taken will also depend on the 'Allow Unearned Discount' and 'Allow Partial Discounts' system parameters. The discount can be manually overridden in the Multiple Invoices Zone of the QuickCash screen.

Can you process non-invoice related receipts through Autolockbox?

No, AutoLockbox is specifically for invoice related receipts. Non-Invoice related receipts, such as investment income, must be processed through the Apply Miscellaneous Transactions form.

Can one customer pay for another customer's invoices through Autolockbox?

Yes, simply list both customers' invoices on the receipt or overflow record. Note, one customer should be identified by a customer number or MICR number on the receipt record. It is not recommended to use AutoAssociate when applying receipts in this way. If your lockbox or system parameter does not allow payment of unrelated invoices, you must first set up a relationship between the customers if you wish to make applications in this way.

How could trimming cause my receipts to display as unidentified?

Receipts are identified by a customer number or MICR number being passed as part of the bank record. They can also be identified by the invoice number when AutoAssociate is used. If this information is supplied, and most of the receipts still show as unidentified, it is usually a problem with how the customer number, MICR number, or invoice number is being trimmed during validation. Trimming is done to remove blanks or zeros used to pad data fields from the bank's datafile. Your Transmission Format determines how a field will be trimmed. You must specify whether the field is right or left justified, and then identify the trim character to be a zero or blank. If the field is right justified, the validation process trims the fill characters from the

left until it reaches a non-fill character. If the field is left justified, the validation process trims the fill characters from the right until it reaches a non-fill character.

Here are some examples:

Character Field, 10 characters long, Right Justified, Zero Filled

Character Field, 10 characters long, Left Justified, Zero Filled

Incorrect trimming can cause a receipt to be unidentified, because an incorrectly trimmed field will not match the corresponding database field during validation. For example, if the customer number should appear as 00842 after validation, but is appearing as 842, it will not match customer number 00842 in your Oracle Receivables application. The trim specifications in the above example are "right justified and zero filled", because the leading zeros are being trimmed until a non-fill character (8) is encountered. To have the customer number appear as 00842 after validation the user can modify the fill character to be "blank" and the leading zeros will not be trimmed.

When does Autolockbox consider a receipt to be a duplicate?

Duplicate receipts have the same receipt number, amount, currency, and customer number. Autolockbox does not allow duplicate receipts within the same batch for the same customer. This is the same validation that is performed when you manually enter receipts into your Oracle Receivables application.

When does Autolockbox consider an invoice to be a duplicate?

Invoice numbers are only required to be unique within a batch source. A customer can have duplicate invoice numbers as long as they belong to different batch sources. (If you have duplicate invoice numbers within a batch source, Autolockbox will reject the application because it will not be able to determine which invoice to apply the receipt against.) You will be able to apply the receipt in the QuickCash or multiple Invoices screen.

What causes an application to be invalid?

Sometimes the Autolockbox Execution Report will show receipts rejected with error code 43281: Receipt has invalid applications. Your application is invalid if:

- 1) The receivable item is not in the same currency as the receipt.

- 2) The receivable item does not belong to the parent or child of the receipt, and Pay Unrelated Invoices is set to No.
- 3) The receivable item is not an invoice, a debit memo, a deposit, a credit memo, a chargeback, or an on-account credit?
- 4) The receivable item is a duplicate or invalid for the customer.
- 5) The receivable item has been selected for automatic receipt.
- 6) The installment number or the receivable item is invalid.

Autolockbox uses the same reasons to invalidate an application as the standard receipt entry forms.

How does Autolockbox divide receipts into batches?

Autolockbox uses 4 criteria for dividing receipts into batches. They are listed in order of precedence as follows:

- 1) A batch can only have one deposit date or GL date. So, if Autolockbox encounters a change in the deposit date or the GL date it will create a new receipt batch.
- 2) A batch can have only one batch name. So, if a new batch name is encountered, it will create a new receipt batch.
- 3) You can specify the maximum size of a batch in the Define Lockboxes form. If the number of receipts exceeds this maximum, Autolockbox will create a new receipt batch.
- 4) The bank can provide batch records as part of the data file, which divide the receipts into batches.

If the group of receipts has one deposit date, GL date and batch name, is less than the maximum size of a batch, and there are no batch records in the data file, they will be processed as one batch.

Integrating Customer Information Using Oracle Receivables Applications Customer Interface

Use Customer Interface to transfer and validate customer, customer profile, contact phone data, payment method data, and customer bank data from other systems. Once customer information is imported into Oracle Receivables and Oracle Government Receivables you can manually update and add new information through the customer entry forms.

Basic Needs

Your Oracle Receivables application provides you with the features you need to satisfy the following basic integration needs. You should be able to:

- Import current or historical customer data such as customer names, addresses, site uses, contacts and telephone numbers. You can also import customer profile class, payment method, and bank information.
- Review the results of your customer data import in a clear and concise report. Identify all successfully imported data as well as data that failed the import process.

Major Features

Simple Integration

Customer Interface lets you import customer data from any feeder system.

Historical Data Conversion

Customer Interface lets you import data from your original system, making the transition from your old application to your Oracle Receivables application as smooth and as efficient as possible. Your Oracle Receivables application does not however, import territory flexfield information.

Complete Validation

Customer Interface ensures that the data which you are importing is accurate and valid within your Oracle Receivables application.

Customer Interface Transfer Report

The Customer Interface Transfer Report is generated every time you run Customer Interface. You can use this report to see the number of customers, addresses, site uses, customer profiles, contacts, telephone numbers, customer bank branches, customer bank accounts and payment methods that were imported into your Oracle Receivables application. This report also displays an exception section to inform you of records that fail validation.

Customer Credit Profiles

Customer Interface handles the update or insert of customer credit profile information. You can assign payment term, statement, automatic receipt, finance charge, dunning and invoicing information to your customers and their Bill-To business purposes.

Automatic Receipts

Customer Interface passes payment methods and banks that are associated with your customers. Your Oracle Receivables application uses this information for the creation of automatic receipts. If you associate an automatic payment method to your customers or their Bill-To business purposes, you must also enter a bank account for this customer.

Definitions

Feeder Program

A custom program you write to transfer customer data from an external system into your Oracle Receivables application interface tables. The type of feeder program you write depends on the environment from which you are importing data.

Customer Business Purpose

The function of a particular location for a customer. For example, your customer can have different billing and shipping locations. Business Purposes are associated with customer addresses.

Customer Contact

A specific customer employee with whom you communicate. Your Oracle Receivables application lets you define as many contacts as you wish for each customer. You can also define contacts for an address and assign previously defined contacts to each business purpose.

Customer Relationships

A relationship created between customers to control receipt and invoice activity. If you create a relationship between two customers each customer can create invoices against commitments belonging to the customer as well as apply receipts to the other customers' items.

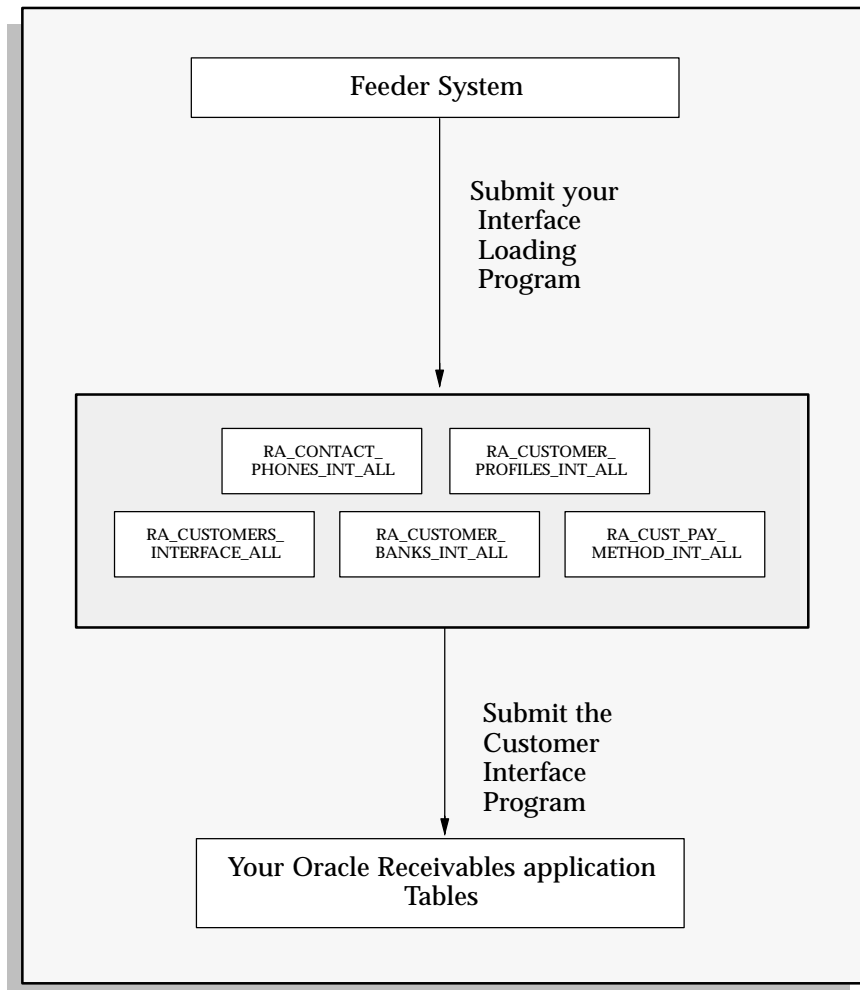
Customer Profiles

A way to categorize your customers based on credit, payment term, statement, automatic receipt, finance charge, dunning and invoicing information. Credit profiles provide default values to speed customer entry.

Overview

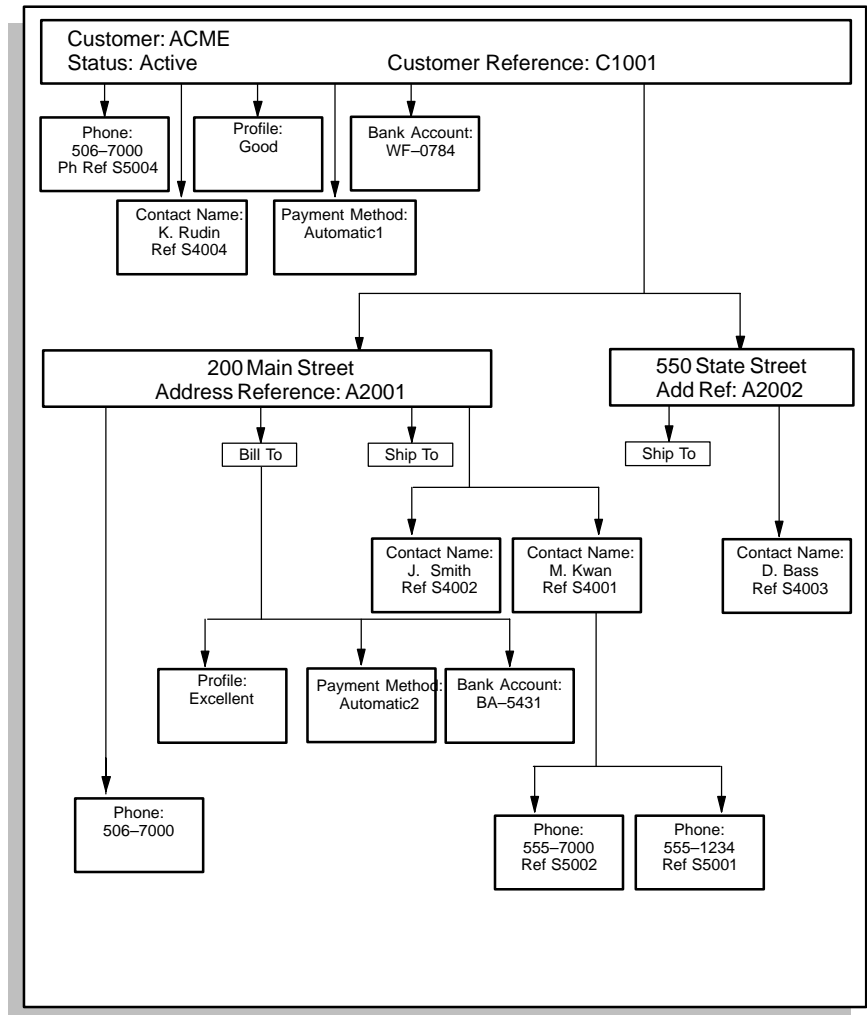
The following diagram shows how customer information is imported into your Oracle Receivables application customer tables.

Figure 10 – 6
Customer Interface



Sample Customer Import The following diagram shows an example of a customer with several addresses, customer profiles, contacts, telephone numbers, business purposes, bank accounts and payment methods. Compare this diagram to the data examples on the next page to see how you would prepare your interface to successfully import this customer information.

Figure 10 – 7 Sample Customer Import



The following example shows you the information that your feeder program needs to load into your Oracle Receivables application Customer Interface tables, to import customer ACME, illustrated on the previous page.

RA_CUSTOMERS_INTERFACE_ALL

Customer Reference	Name	Address Reference	Address	Site Code
C1001	ACME	A2001	200 Main	BILL_TO
C1001	ACME	A2001	200 Main	SHIP_TO
C1001	ACME	A2002	550 State	SHIP_TO

Table 10 – 10 (Page 1 of 1)

RA_CUSTOMER_PROFILES_INT_ALL

Customer Reference	Address Reference	Profile Name
C1001		Good
C1001	A2001 (This address reference refers to the Bill-To site)	Excellent

Table 10 – 11 (Page 1 of 1)

RA_CONTACT_PHONES_INT_ALL

Customer Reference	Address Reference	Contact Reference	Last Name	Phone Reference	Phone Number
C1001	A2001	S4001	KWAN	S5001	555-1234
C1001	A2001	S4001	KWAN	S5002	555-7000
C1001	A2001	S4002	SMITH		
C1001	A2001			S5003	474-8664
C1001	A2002	S4003	BASS		
C1001		S4004	RUDIN		
C1001				S5004	506-7000

Table 10 – 12 (Page 1 of 1)

RA_CUSTOMER_BANKS_INT_ALL

Customer Reference	Address Reference	Bank Account
C1001		WF-0784
C1001	A2001 (This address reference refers to the Bill-To site)	BA-5431

Table 10 – 13 (Page 1 of 1)

RA_CUST_PAY_METHOD_INTERFACE

Customer Reference	Address Reference	Payment Method
C1001		Automatic1
C1001	A2001 (This address reference refers to the Bill-To site)	Automatic2

Table 10 – 14 (Page 1 of 1)

Preparing Oracle Receivables and Oracle Government Receivables

To ensure that Customer Interface runs smoothly, you need to prepare your Oracle Receivables application for any new data that you require Customer Interface to import.

- QuickCodes
 - Countries
 - Site Use Codes
 - Credit ratings
 - Risk Codes
 - Account Statuses
 - Communication Types
 - Customer Classes
- Freight Carriers

- Demand Classes
- AutoCash Rule Sets
- Payment Terms
- AutoInvoice Grouping Rules
- Collectors
- Payment Methods
- Statement Cycles
- Dunning Letter Sets
- Customer Profile Classes
- Customer Bank Information
- Tax Codes
- Customer Exemptions



Setting Up Oracle Receivables and Oracle Government
Receivables
*Oracle Financials and Oracle Government Financials
Implementation Manual*

Importing Data From Your Feeder System

Customer Interface receives data from an import program which is used to convert data from your feeder system into a standard data format that Customer Interface can read. Customer Interface can then convert your import data into your Oracle Receivables application customer information.

Write an Import Program

You need to write an import program that is compatible with the type of environment from which you want to import your data. For example, you can use SQL*Loader, SQL*Report, PL/SQL, or C to write an import program to import data from a non-Oracle system. You can also write a conversion program to import historical data from your original customer database. Regardless of the type of import program you write, the output should be in standard data format that Customer Interface can use to import into your Oracle Receivables application.

Reference Tables

The following tables display all of the required columns for each Customer Interface table. Use this information to help you determine what information you must pass. For example, if you want to enter a new contact for a previously entered customer, you must enter values for ORIG_SYSTEM_CUSTOMER_REF, ORIG_SYSTEM_CONTACT_REF, INSERT_UPDATE_FLAG and CONTACT_LAST_NAME. Only the required columns are listed. Customer Interface lets you pass additional information into non-required columns if you wish.

RA_CUSTOMERS_INTERFACE_ALL

Customers	Customers, Addresses and Business Purposes
ORIG_SYSTEM_CUSTOMER_REF	ORIG_SYSTEM_CUSTOMER_REF
INSERT_UPDATE_FLAG	INSERT_UPDATE_FLAG
CUSTOMER_NAME	CUSTOMER_NAME
CUSTOMER_NUMBER (If you are not using Automatic Customer Numbering)	CUSTOMER_NUMBER (If you are not using Automatic Customer Numbering)
CUSTOMER_STATUS	CUSTOMER_STATUS
A customer level profile must exist in RA_CUSTOMER_PROFILES_INT_ALL for new customers	A customer level profile must exist in RA_CUSTOMER_PROFILES_INT_ALL for new customers
	PRIMARY_SITE_USE_FLAG
	LOCATION (If you are not using Automatic Site Numbering)
	SITE_USE_CODE
	ADDRESS1
	ORIG_SYSTEM_ADDRESS_REF
	You must enter values for the columns you reference in your Tax Location Flexfield if you are calculating sales tax and Address Validation is set to 'Error' CITY, STATE, PROVINCE, COUNTY, POSTAL_CODE and COUNTRY
	COUNTRY

Table 10 – 15 RA_CUSTOMERS_INTERFACE_ALL (Page 1 of 2)

Customers	Customers, Addresses and Business Purposes
LAST_UPDATED_BY	LAST_UPDATED_BY
LAST_UPDATE_DATE	LAST_UPDATE_DATE
CREATED_BY	CREATED_BY
CREATION_DATE	CREATION_DATE

Table 10 – 15 RA_CUSTOMERS_INTERFACE_ALL (Page 2 of 2)

RA_CUSTOMER_PROFILES_INT_ALL

Customer Profiles for Customers	Customer Profiles for Addresses
ORIG_SYSTEM_CUSTOMER_REF	ORIG_SYSTEM_CUSTOMER_REF
INSERT_UPDATE_FLAG	INSERT_UPDATE_FLAG
CUSTOMER_PROFILE_CLASS_NAME (If you did not pass a value in this column you must enter values in: COLLECTOR_NAME CREDIT_BALANCE_STATEMENTS CREDIT_CHECKING AUTO_REC_INCL_DISPUTED_FLAG DISCOUNT_TERMS DUNNING_LETTERS (if this column is 'Y' you must also enter a value in DUN- NING_LETTER_SET_NAME) INTEREST_CHARGES (If 'Y' you must also enter values in INTEREST_ PERIOD_DAYS and CHARGE_ON_ FINANCE_CHARGE_FLAG) STATEMENTS (If this column is 'Y' you must also enter a value in STATEMENT_CYCLE_NAME) TOLERANCE TAX_PRINTING_OPTION OVERRIDE_TERMS GROUPING_RULE_NAME	CUSTOMER_PROFILE_CLASS_NAME (If you did not pass a value in this column you must enter values in: COLLECTOR_NAME CREDIT_BALANCE_STATEMENTS CREDIT_CHECKING AUTO_REC_INCL_DISPUTED_FLAG DISCOUNT_TERMS DUNNING_LETTERS (if this column is 'Y' you must also enter a value in DUN- NING_LETTER_SET_NAME) INTEREST_CHARGES (If 'Y' you must also enter values in INTEREST_ PERIOD_DAYS and CHARGE_ON_ FINANCE_CHARGE_FLAG) STATEMENTS (If this column is 'Y' you must also enter a value in STATEMENT_CYCLE_NAME) TOLERANCE TAX_PRINTING_OPTION OVERRIDE_TERMS GROUPING_RULE_NAME
	ORIG_SYSTEM_ADDRESS_REF (must reference a Bill-To site)
CREDIT_HOLD	CREDIT_HOLD

Table 10 – 16 RA_CUSTOMER_PROFILES_INT_ALL (Page 1 of 2)

Customer Profiles for Customers	Customer Profiles for Addresses
LAST_UPDATED_BY	LAST_UPDATED_BY
LAST_UPDATE_DATE	LAST_UPDATE_DATE
CREATED_BY	CREATED_BY
CREATION_DATE	CREATION_DATE

Table 10 – 16 RA_CUSTOMER_PROFILES_INT_ALL (Page 2 of 2)

RA_CONTACT_PHONES_INT_ALL

Telephone Numbers for Customers	Telephone Numbers for Addresses	Telephone Numbers for Contacts
ORIG_SYSTEM_CUSTOMER_REF	ORIG_SYSTEM_CUSTOMER_REF	ORIG_SYSTEM_CUSTOMER_REF
ORIG_SYSTEM_TELEPHONE_REF	ORIG_SYSTEM_TELEPHONE_REF	ORIG_SYSTEM_TELEPHONE_REF
INSERT_UPDATE_FLAG	INSERT_UPDATE_FLAG	INSERT_UPDATE_FLAG
TELEPHONE	TELEPHONE	TELEPHONE
TELEPHONE_TYPE	TELEPHONE_TYPE	TELEPHONE_TYPE
	ORIG_SYSTEM_ADDRESS_REF	ORIG_SYSTEM_CONTACT_REF
		CONTACT_LAST_NAME
LAST_UPDATED_BY	LAST_UPDATED_BY	LAST_UPDATED_BY
LAST_UPDATE_DATE	LAST_UPDATE_DATE	LAST_UPDATE_DATE
CREATED_BY	CREATED_BY	CREATED_BY
CREATION_DATE	CREATION_DATE	CREATION_DATE

Table 10 – 17 RA_CONTACT_PHONES_INT_ALL (Page 1 of 1)

RA_CONTACT_PHONES_INT_ALL

Contacts for Customers	Contacts for Addresses
ORIG_SYSTEM_CUSTOMER_REF	ORIG_SYSTEM_CUSTOMER_REF
ORIG_SYSTEM_CONTACT_REF	ORIG_SYSTEM_CONTACT_REF

Contacts for Customers	Contacts for Addresses
INSERT_UPDATE_FLAG	INSERT_UPDATE_FLAG
CONTACT_LAST_NAME	CONTACT_LAST_NAME
	ORIG_SYSTEM_ADDRESS_REF
LAST_UPDATED_BY	LAST_UPDATED_BY
LAST_UPDATE_DATE	LAST_UPDATE_DATE
CREATED_BY	CREATED_BY
CREATION_DATE	CREATION_DATE

Table 10 – 18 RA_CONTACT_PHONES_INT_ALL (Page 2 of 2)

RA_CUSTOMER_BANKS_INT_ALL

Banks for Customers	Banks for Bill-To Addresses
ORIG_SYSTEM_CUSTOMER_REF	ORIG_SYSTEM_CUSTOMER_REF
PRIMARY_FLAG	PRIMARY_FLAG
START_DATE	START_DATE
BANK_ACCOUNT_NAME (If the bank account does not exist you must also enter values in: BANK_ACCOUNT_CURRENCY_CODE BANK_ACCOUNT_NUM BANK_ACCOUNT_NAME BANK_BRANCH_NAME	BANK_ACCOUNT_NAME (If the bank account does not exist you must also enter values in: BANK_ACCOUNT_CURRENCY_CODE BANK_ACCOUNT_NUM BANK_ACCOUNT_NAME BANK_BRANCH_NAME
	ORIG_SYSTEM_ADDRESS_REF (Must reference a Bill-To site)
LAST_UPDATED_BY	LAST_UPDATED_BY
LAST_UPDATE_DATE	LAST_UPDATE_DATE
CREATED_BY	CREATED_BY
CREATION_DATE	CREATION_DATE

Table 10 – 19 RA_CUSTOMER_BANKS_INT_ALL (Page 1 of 1)

RA_CUST_PAY_METHOD_INTERFACE

Payment Methods for Customers	Payment Methods for Bill-To Addresses
ORIG_SYSTEM_CUSTOMER_REF	ORIG_SYSTEM_CUSTOMER_REF
PAYMENT_METHOD_NAME	PAYMENT_METHOD_NAME
PRIMARY_FLAG	PRIMARY_FLAG
START_DATE	START_DATE
	ORIG_SYSTEM_ADDRESS_REF (Must reference a Bill-To site)
LAST_UPDATED_BY	LAST_UPDATED_BY
LAST_UPDATE_DATE	LAST_UPDATE_DATE
CREATED_BY	CREATED_BY
CREATION_DATE	CREATION_DATE

Table 10 – 20 RA_CUST_PAY_METHOD_INTERFACE (Page 1 of 1)

Customer Interface Validation

The Customer Interface program validates your data for compatibility with your Oracle Receivables application. It ensures that the columns in your Oracle Receivables application interface tables reference the appropriate values and columns in your Oracle Receivables application.

Existence

For some columns Customer Interface ensures that the values are already defined in your Oracle Receivables application.

If you are trying to perform updates, Customer Interface ensures that the record which is to be updated either exists in your Oracle Receivables application or is sitting in the interface table in the insert mode.

Cross Validation

Customer Interface ensures that certain column values are consistent with each other. Thus if no profile class is assigned to a customer, it ensures that interest charges, collector name, discount terms etc. are defined in their appropriate columns.

Uniqueness

Customer Interface ensures that records marked for insert are unique. Thus it will not allow you to insert a customer, profile, bank etc. which already exists in your Oracle Receivables application.

Caution: When loading the interface tables you should remove all trailing spaces from the import data. Otherwise, if you attempt to load two records with same customer name, but one of the records has trailing spaces, Customer Interface will treat each record as unique. If you are using SQL*loader to load the interface tables, you can easily remove all trailing spaces from the import data.

Using Customer Interface

Running Customer Interface

Running Customer Interface can be a one-step process, depending on the accuracy of your feeder program. If your feeder program transfers customer information from other sources into the required data format, and all of the data passes validation, you can run Customer Interface successfully in one step.

Run Customer Interface using the Run Customer Interface form in your Oracle Receivables application. After Customer Interface has successfully imported customer data into your Oracle Receivables application you can use the Enter Customer Information and Quick Customer Entry forms to maintain this information.



Enter Customer Information

Quick Customer Entry

(*Oracle Receivables Reference Manual* or *Oracle Government Receivables Reference Manual*)

Customer Interface Validation

Customer Interface validates your data by ensuring that the data in your Oracle Receivables application interface tables correctly reference existing values in your Oracle Receivables application the database.

Customer Interface Transfer Report

The Customer Interface Transfer Report provides multiple error messages per interface record to help you identify and fix problems. You may need to make changes in either Oracle Receivables and Oracle

Government Receivables or your feeder program. For example, if you receive an error message explaining that the country that you specified for an address does not exist in your Oracle Receivables application, then you can either enter this country in the Maintain QuickCode form, or fix your feeder program to import only countries that your Oracle Receivables application recognizes.

CITY
STATE
PROVINCE
COUNTY
POSTAL_CODE
COUNTRY

Enter the city, state, province, county, postal code or country for this customer's address. You must enter values that have already been defined in AR_LOCATION_VALUES if Address Validation is set to 'Error' and you are calculating sales tax. You must do this for each address component that your tax location flexfield structure is based on.

Oracle Receivables and Oracle Government Receivables Tables

Customer Interface transfers customer data into the following your Oracle Receivables application tables:

AR_CUSTOMER_PROFILES
AR_CUSTOMER_PROFILE_AMOUNTS
RA_ADDRESSES
RA_CONTACTS
RA_CUSTOMERS
RA_CUSTOMER_RELATIONSHIPS
RA_CUST_RECEIPT_METHODS
RA_PHONES
RA_SITE_USES
AP_BANK_ACCOUNT_USES
AP_BANK_ACCOUNTS
AP_BANK_BRANCHES

No updates will be allowed to be made through the customer interface to the following tables:

RA_SITE_USES
RA_CUSTOMER_RELATIONSHIPS
RA_CUST_RECEIPT_METHODS

AP_BANK_BRANCHES
AP_BANK_ACCOUNTS
AP_BANK_ACCOUNT_USES

The Customer Interface program will not flag errors against records in the interface table that attempt to update the above tables, however no updates will be performed.

Table and Column Descriptions

Below is a detailed description of the five interface tables into which you load customer data from your feeder system.

RA_CUSTOMERS_INTERFACE_ALL

This table stores customer, address and business purpose information. You do not have to enter values in this table if you do not want to import customers, addresses or business purposes.

Column Name	Value
ADDRESS1 through 4	<p>Enter the address for your customer in these four columns. You can enter up to four lines of an address.</p> <p>Validation: If you enter a value in ORIG_SYSTEM_ADDRESS_REF, you must enter a value in ADDRESS1. Multiple rows with same address reference must have some values in address 1-4.</p> <p>Destination: RA_ADDRESSES.ADDRESS1, RA_ADDRESSES.ADDRESS2, RA_ADDRESSES.ADDRESS3, RA_ADDRESSES.ADDRESS4</p>
CITY STATE PROVINCE COUNTY POSTAL_CODE COUNTRY	<p>Enter the city, state, province, county, postal code or country for this customer's address. You must enter values that have already been defined in AR_LOCATION_VALUES if Address Validation is set to 'Error' and you are calculating sales tax. You must do this for each address component that your tax location flexfield structure is based on.</p>

If Address Validation is set to 'Warning' and you pass an address component value that does not exist in AR_LOCATION_VALUES, Customer Interface inserts this value into AR_LOCATION_VALUES and displays a warning message in the Customer Interface Transfer report.

If Address Validation is set to 'No Validation' and you pass a value that does not exist in AR_LOCATION_VALUES, Customer Interface inserts this value into AR_LOCATION_VALUES.

Validation: The county must exist in FND_TERRITORIES

Destination: RA_ADDRESSES.CITY
RA_ADDRESSES.STATE
RA_ADDRESSES.PROVINCE
RA_ADDRESSES.COUNTY
RA_ADDRESSES.POSTAL_CODE
RA_ADDRESSES.COUNTRY

ADDRESS_KEY

This column is not currently used by Customer Interface.

BILL_TO_ORIG_ ADDRESS_REF

Enter the Bill-To location that you want to associate with the Ship-To address on this record.

Validation: You can enter a value only if the SITE_USE_CODE column in this record is 'Ship-To'. Also the Bill-To address must exist for this or any of the related customers.

Destination: RA_SITE_USES.BILL_TO_SITE_USE_ID

**CUSTOMER_
ATTRIBUTE_
CATEGORY**

Enter Descriptive Flexfield category information. These columns are optional.

**ADDRESS_
ATTRIBUTE_
CATEGORY**

**SITE_USE_
ATTRIBUTE_
CATEGORY**

Validation: None
Destination: RA_CUSTOMERS.ATTRIBUTE_CATEGORY
RA_ADDRESSES.ATTRIBUTE_CATEGORY
RA_SITE_USES.ATTRIBUTE_CATEGORY

**CUSTOMER_
ATTRIBUTE_1 to 15**

Enter Descriptive Flexfield information. These columns are optional.

**ADDRESS_
ATTRIBUTE_1 to 15**

**SITE_USE_
ATTRIBUTE_1 to 25**

Validation: None
Destination: RA_CUSTOMERS.ATTRIBUTE1 to 15
RA_ADDRESSES.ATTRIBUTE1 to 15
RA_SITE_USES.ATTRIBUTE1 to 25

**CUSTOMER_
CATEGORY_CODE**

Enter either 'Customer' or 'Prospect' to categorize your customer. This column is required.

Validation: AR_LOOKUPS.LOOKUP_CODE where
LOOKUP_TYPE = 'CUSTOMER_CATEGORY'
Destination: RA_CUSTOMERS.CUSTOMER_CATEGORY_
CODE

**CUSTOMER_
CLASS_CODE**

Enter the customer class for this customer. You can use customer classes to categorize your customers by their business type, size or location. Use customer classes that you previously defined in the Define QuickCodes form with a QuickCode type of 'Customer Class'. This column is optional.

	Validation: AR_LOOKUPS.LOOKUP_CODE where LOOKUP_TYPE = 'CUSTOMER CLASS' Destination: RA_CUSTOMERS.CUSTOMER_CLASS
CUSTOMER_KEY	This column is not currently used by Customer Interface.
CUSTOMER_NAME	Enter the name of your customer. This column is required. Validation: The same customer reference cannot have different customer names within this table. Destination: RA_CUSTOMERS.CUSTOMER_NAME
CUSTOMER_ NUMBER	Enter this customer's number. Validation: Must be null if you are using Automatic Customer Numbering. Must exist if you are not using Automatic Customer Numbering. This value must be unique within RA_CUSTOMERS. Destination: RA_CUSTOMERS.CUSTOMER_NUMBER
CUSTOMER_STATUS	Enter the status for this customer. This column is required. Validation: Must equal 'A' for Active or 'I' for Inactive Destination: RA_CUSTOMERS.STATUS
CUSTOMER_TYPE	Enter 'Internal' or 'External' to indicate customer type for this customer. Choices include Internal and External. This column is optional. Validation: AR_LOOKUPS.LOOKUP_CODE where LOOKUP_TYPE = 'CUSTOMER_TYPE' Multiple rows with the same customer reference must have the same customer type.
CUST_TAX_CODE	Enter the tax code that you want to assign to this customer or business purpose, depending on the column you choose. You must use tax codes that you previously defined in the Define Tax Codes and Rates form. These columns are optional.
SITE_USE_TAX_ CODE	Validation: Must exist in AR_VAT_TAX Destination: RA_CUSTOMERS.TAX_CODE RA_SITE_USES.TAX_CODE

CUST_TAX_ EXEMPT_NUM	<p>This column is not currently used by Customer Interface and must be left blank.</p> <p>Validation: None</p> <p>Destination: None</p>
CUST_TAX_ REFERENCE	<p>Enter the tax registration number for this customer.</p> <p>Validation: None</p> <p>Destination: RA_CUSTOMERS.TAX_REFERENCE</p>
CUST_SHIP_ VIA_CODE	<p>Enter the freight carrier you want to assign to this customer or business purpose, depending on the column you choose. Use freight carriers that you previously defined in the Define Freight Carrier form. This column is optional.</p> <p>Validation: Must exist in ORG_FREIGHT</p> <p>Destination: RA_CUSTOMERS.SHIP_VIA RA_SITE_USES.SHIP_VIA</p>
SITE_SHIP_ VIA_CODE	
CREATED_BY	<p>Enter the user id that is creating this row. This column is required.</p> <p>Validation: None</p> <p>Destination: None</p>
CREATION_DATE	<p>Enter the system date. This column is required.</p> <p>Validation: None</p> <p>Destination: None</p>
DEMAND_ CLASS_CODE	<p>Enter the demand class for this address. Use demand classes that you previously defined in the Define Demand Classes form. This column is optional.</p> <p>Validation: Must exist in FND_COMMON_LOOKUPS</p> <p>Destination: RA_SITE_USES.DEMAND_CLASS_CODE</p>
INSERT_UPDATE_ FLAG	<p>Specify whether you are inserting a new record or updating an existing record. This column is required.</p> <p>Validation: 'I' for insert or 'U' for update</p> <p>Destination: None</p>
INTERFACE_STATUS	<p>This column is used by Customer Interface and should be left null. The Customer Interface program updates this column with all error messages that apply to this interface record. If an interface record has</p>

several problems, the Customer Interface program updates this column with multiple error codes.

LOCATION

Enter a short hand name for this business purpose. You use this value to quickly refer to a business purpose during data entry.

Validation: If AR_SYSTEM_PARAMETERS.AUTO_SITE_NUMBERING is set to 'N', you must enter a value in this column. If AR_SYSTEM_PARAMETERS.AUTO_SITE_NUMBERING is set to 'Y', do not enter a value. Inserts for this column must be unique.

Destination: RA_SITE_USES.LOCATION

LOCATION_CCID

This column is used by Customer Interface and should be left null. Customer Interface stores the code combination id of valid addresses in this column.

Validation: None

Destination: None

LAST_UPDATED_BY

Enter the user id that is updating this row. This column is required.

Validation: None

Destination: None

LAST_UPDATE_DATE

Enter the system date. This column is required.

Validation: None

Destination: None

LAST_UPDATE_LOGIN

Enter the login id. This column is optional

Validation: None

Destination: None

LANGUAGE

Enter the language used by this customer site.

Validation: Must exist in FND_LANGUAGES.NLS_LANGUAGE

Destination: RA_ADDRESSES.LANGUAGE

MESSAGE_TEXT

This column is used by Customer Interface and should be left null.

Validation: None

Destination: None

ORG_ID

If you use the Oracle Applications multiple organizations feature, enter the operating unit's org_id. If you do not use the Oracle Applications multiple organizations feature, leave this column null. To insert rows into open interface tables, use the multiple organization table RA_CUSTOMERS_INTERFACE_ALL and explicitly populate the org_id column. You can determine an operating unit's org_id by querying the HR_OPERATING_UNITS view:

```
SELECT name, organization_id from  
HR_OPERATING_UNITS;
```

When you import rows *from* open interface tables, only rows from your current operating unit are processed.

ORIG_SYSTEM_ ADDRESS_REF

Enter a value you can use to uniquely identify this address in your original system. The reference value that you enter provides you with an audit trail from your Oracle Receivables application back to your original system.

This column forms part of the primary key for RA_CUSTOMERS_INTERFACE_ALL. The primary key is a combination of ORIG_SYSTEM_CUSTOMER_REF, ORIG_SYSTEM_ADDRESS_REF, and SITE_USE_CODE.

To enter multiple addresses for a customer, enter multiple records in RA_CUSTOMERS_INTERFACE_ALL with identical customer information, but with different address information.

This column is required if you are either inserting or updating address information.

Validation: Must not already exist in RA_ADDRESSES for insert.
Must already exist in RA_ADDRESSES for update

Destination: RA_ADDRESSES.ORIG_SYSTEM_REFERENCE



Suggestion: If the value you enter is numeric, add an alpha character to the end to ensure that this number never conflicts with a system-generated ID number.

ORIG_SYSTEM_ CUSTOMER_REF

Enter a value which uniquely identifies this customer in your original system. The reference number you enter here provides you with an audit trail from your Oracle Receivables application back to your original system.

This column forms part of the primary key for RA_CUSTOMERS_INTERFACE_ALL. The primary key is a

combination of ORIG_SYSTEM_CUSTOMER_REF,
ORIG_SYSTEM_ADDRESS_REF, and SITE_USE_CODE.

If you are entering a new customer, you must also enter a customer level profile in RA_CUSTOMER_PROFILES_INT_ALL. This column is required.

Validation: Must not exist in RA_CUSTOMERS for insert.
Must exist in RA_CUSTOMERS for update
The same customer reference cannot have different customer names within this table. Inserts for this column must be unique.

Destination: RA_CUSTOMERS.ORIG_SYSTEM_REFERENCE



Suggestion: If the value you enter is numeric, add an alpha character to the end to ensure that this number never conflicts with a system-generated ID number.

ORIG_SYSTEM_ PARENT_REF

Enter the original system reference of the related customer, if there is one. If you enter a value in this column, your Oracle Receivables application checks

RA_SYSTEM_PARAMETERS.CREATE_RECIPROCAL_FLAG to determine whether the application should automatically create the reciprocal relationship.

If RA_SYSTEM_PARAMETERS.CREATE_RECIPROCAL_FLAG is set to 'Y', your Oracle Receivables application creates an additional, opposite entry in RA_CUSTOMER_RELATIONSHIPS. This column is optional.

Validation: Must exist in RA_CUSTOMERS.
ORIG_SYSTEM_REFERENCE
Multiple rows with the same customer reference must have the same
ORIG_SYSTEM_PARENT_REF.

Destination: Inserts into RA_CUSTOMER_RELATIONSHIPS.
CUSTOMER_ID select CUSTOMER_ID from
RA_CUSTOMERS where ORIG_SYSTEM_
REFERENCE =
RA_CUSTOMERS_INTERFACE_ALL.
ORIG_SYSTEM_PARENT_REF

PRIMARY_SITE_ USE_FLAG

Enter 'Y' or 'N' to indicate whether this is the primary business purpose. Only enter a value in this column if the INSERT_UPDATE_FLAG is 'Y' and you enter a value in ORIG_SYSTEM_ADDRESS_REF.

Validation: Must only have one primary business purpose for each usage (Bill-To, Ship-To, etc.)
Must be null, 'Y' or 'N'.
Mandatory when inserting an address and must be null when the record is for updating purposes.
Not updatable

Destination: RA_SITE_USES.PRIMARY_FLAG

REQUEST_ID

This column is used by Customer Interface and should be left null.

SITE_USE_CODE

Enter the business purpose for this customer's address. Use business purposes you previously define in the Maintain QuickCodes form with a QuickCode type of 'Business purposes for a customer address'.

This column forms part of the primary key for RA_CUSTOMERS_INTERFACE_ALL. The primary key is a combination of ORIG_SYSTEM_CUSTOMER_REF, ORIG_SYSTEM_ADDRESS_REF, and SITE_USE_CODE.

If you enter a value in ORIG_SYSTEM_ADDRESS_REF, you must enter a value in this column. To enter multiple business purposes for an address, enter multiple records in RA_CUSTOMERS_INTERFACE_ALL with identical customer and address information, but with different site uses. You can only assign one type of business purpose to each address.

Validation: Must equal a value in AR_LOOKUPS.
LOOKUP_CODE where LOOKUP_TYPE = 'SITE_USE_CODE'
This column must be null if you are updating PRIMARY_SITE_USE_FLAG, LOCATION, SITE_SHIP_VIA_CODE, SITE_USE_TAX_CODE or SITE_USE_TAX_REFERENCE. Inserts for this column must be unique.

Destination: RA_SITE_USES.SITE_USE_CODE

SITE_USE_TAX_EXEMPT_NUM

This column is not currently used by Customer Interface and must be left blank.

Validation: None

Destination: None

SITE_USE_TAX_REFERENCE

Enter the tax registration number for this site.

Validation: None

	Destination: RA_SITE_USES.TAX_REFERENCE
VALIDATED_FLAG	This column is used by Customer Interface and should be left null.
	Destination: RA_CUSTOMERS.CUSTOMER_TYPE
WARNING_TEXT	This column is not currently used by Customer Interface.

RA_CUSTOMER_PROFILES_INT_ALL

This table stores customer profile information. If you are entering a new customer you must either pass a customer profile class that already exists in your Oracle Receivables application or customer profile values. You do not have to enter values in this table if you are not entering a new customer or assigning customer profile information to customer addresses.

Column Name	Value
ACCOUNT_STATUS	Enter the status of this customer's account. Use account statuses you previously defined in the Maintain QuickCodes form with a QuickCode type of 'Account Status'. This column is optional. Validation: AR_LOOKUPS.LOOKUP_CODE where LOOKUP_TYPE = 'ACCOUNT_STATUS' Destination: AR_CUSTOMER_PROFILES.ACCOUNT_STATUS
ATTRIBUTE_CATEGORY	Enter Descriptive Flexfield category information. These columns are optional. Validation: None Destination: AR_CUSTOMER_PROFILES.ATTRIBUTE_CATEGORY AR_CUSTOMER_PROFILE_AMOUNTS.ATTRIBUTE_CATEGORY
ATTRIBUTE 1 through 15	Enter Descriptive Flexfield information. These columns are optional. Validation: None Destination: AR_CUSTOMER_PROFILES.ATTRIBUTE1 through 15 AR_CUSTOMER_PROFILE_AMOUNTS.ATTRIBUTE1 through 15

AUTO_REC_INCL_DISPUTED_FLAG Specify whether you want to include debit items that have been placed in dispute when you create automatic receipts for your customers. Defaults to No if null. This column is optional.

Validation: Must equal 'Y' (Yes) or 'N' (No)
Mandatory when profile class is null.

Destination: AR_CUSTOMER_PROFILES.AUTO_REC_INCL_DISPUTED_FLAG

AUTOCASH_HIERARCHY_NAME Enter the AutoCash Rule set that you want to assign to this customer. Use AutoCash Rules sets that you previously defined in the Define AutoCash Rules form. AutoCash Rule sets determine the sequence of AutoCash Rules that your Oracle Receivables application uses when automatically applying receipts to your customer's open debit items. This column is optional.

Validation: Must exist in AR_AUTOCASH_HIERARCHIES
Mandatory when no profile class specified.

Destination: AR_CUSTOMER_PROFILES.
AUTOCASH_HIERARCHY_ID

AUTO_REC_MIN_RECEIPT_AMOUNT Enter the minimum receipt amount that must be specified for this customer when you create automatic receipts in this currency. This column is optional.

Validation: None

Destination: AR_CUSTOMER_PROFILE_AMOUNTS.
AUTO_REC_MIN_RECEIPT_AMOUNT

CHARGE_ON_FINANCE_CHARGE_FLAG Specify whether you want to compound interest for this customer. You must enter a value in this column if INTEREST_CHARGES is set to 'Y' (Yes). Do not enter a value if INTEREST_CHARGES is null or set to 'N' (No).

If you enter a customer profile class in CUSTOMER_PROFILE_CLASS_NAME, your Oracle Receivables application defaults to the value associated with that profile class if this column is null.

If you do not enter a value in CUSTOMER_PROFILE_CLASS_NAME and INTEREST_CHARGES is set to 'Y' (Yes), or you want to override the value associated with the customer profile class, you must enter a value in this column.

	<p>Validation: Must equal 'Y' (Yes) or 'N' (No) Mandatory when INTEREST_CHARGES is Yes Must be null if INTEREST_CHARGES is No or null</p> <p>Destination: AR_CUSTOMER_PROFILES.CHARGE_ON_FINANCE_CHARGE_FLAG</p>
CLEARING_DAYS	<p>Enter the number of clearing days for this customer profile.</p> <p>Validation: Must be an integer greater than or equal to zero.</p> <p>Destination: AR_CUSTOMER_PROFILES.CLEARING_DAYS</p>
COLLECTOR_NAME	<p>Enter the collector assigned to this customer profile.</p> <p>If you enter a customer profile class in CUSTOMER_PROFILE_CLASS_NAME, your Oracle Receivables application defaults to the collector associated with that profile class if this column is null.</p> <p>If you do not enter a value in CUSTOMER_PROFILE_CLASS_NAME, or you want to override the collector associated with the customer profile class, you must enter a value in this column.</p> <p>Validation: Must be unique in AR_COLLECTORS and STATUS = 'A' (Active). Mandatory when no profile class specified.</p> <p>Destination: AR_CUSTOMER_PROFILES.COLLECTOR_ID (derived from CUSTOMER_NAME)</p>
CREATED_BY	<p>Enter the user id that is creating this row. This column is required.</p> <p>Validation: None</p> <p>Destination: None</p>
CREATION_DATE	<p>Enter the system date. This column is required.</p> <p>Validation: None</p> <p>Destination: None</p>
CREDIT_BALANCE_STATEMENTS	<p>Specify whether you want to send statements to customers with credit balances.</p> <p>If you enter a customer profile class in CUSTOMER_PROFILE_CLASS_NAME, your Oracle Receivables application defaults to the value associated with that profile class if this column is null.</p>

If you do not enter a value in CUSTOMER_PROFILE_CLASS_NAME or you want to override the value associated with the customer profile class, you must enter a value in this column.

Validation: Must equal 'Y' (Yes) or 'N' (No)
Must be 'N' when STATEMENTS = 'N'
Mandatory when no profile class specified.
Mandatory when STATEMENTS = Yes
Must be null when STATEMENTS is null.

Destination: AR_CUSTOMER_PROFILES.CREDIT_BALANCE_STATEMENTS

CREDIT_HOLD

Specify whether you want to put a hold on your customer's credit. This column is optional.

Validation: Must equal 'Y' (Yes) or 'N' (No)'

Destination: AR_CUSTOMER_PROFILES.CREDIT_HOLD

CREDIT_RATING

Enter the credit rating for this customer. Use credit ratings you previously defined in the Maintain QuickCodes form with a QuickCode type of 'Credit rating for customers'. This column is optional.

Validation: AR_LOOKUPS.LOOKUP_CODE where
LOOKUP_TYPE = 'CREDIT_RATING'

Destination: AR_CUSTOMER_PROFILES.CREDIT_RATING

CURRENCY_CODE

Enter a currency code to define a customer profile amount for this customer. Use currency codes previously defined in the Define Currency form. Regardless of the value stored in INSERT_UPDATE_FLAG, Customer Interface always inserts and updates customer profile amount values you pass in this table. You must enter a value if any one of the following columns have values:

INTEREST_RATE

MIN_DUNNING_AMOUNT

MIN_DUNNING_INVOICE_AMOUNT

MIN_FC_BALANCE_AMOUNT

MIN_FC_INVOICE_AMOUNT

MIN_STATEMENT_AMOUNT

OVERALL_CREDIT_LIMIT

TRX_CREDIT_LIMIT

Validation: Must exist in FND_CURRENCIES
Mandatory when a profile amount value is populated
(Profile amount columns are listed above)

Destination: AR_CUSTOMER_PROFILE_AMOUNTS.
CURRENCY_CODE

CUSTOMER_PROFILE_CLASS_NAME Enter the name of the customer profile class you want to assign to this customer or Bill-To address.

If you enter a value in this column, your Oracle Receivables application will use the profile values associated with this customer profile class. However, you can override these default values by passing new values in the appropriate columns. This column is optional.

However, if you do not enter a value in this column you must enter values in the following columns:

COLLECTOR_NAME

CREDIT_BALANCE_STATEMENTS

CREDIT_CHECKING

AUTO_REC_INCL_DISPUTED_FLAG

DISCOUNT_TERMS

DUNNING_LETTERS (If 'Y', you must also enter a value in DUNNING_LETTER_SET_NAME)

INTEREST_CHARGES (If 'Y', you must also enter values in INTEREST_PERIOD_DAYS and CHARGE_ON_FINANCE_CHARGE_FLAG)

STATEMENTS (If 'Y', you must also enter a value in STATEMENT_CYCLE_NAME)

TOLERANCE

TAX_PRINTING_OPTION

OVERRIDE_TERMS

GROUPING_RULE_NAME

Validation: Must equal AR_CUSTOMER_PROFILE_CLASSES.
NAME and STATUS = 'A' (Active)

Destination: AR_CUSTOMER_PROFILES.
CUSTOMER_PROFILE_CLASS_ID (Derived from
CUSTOMER_PROFILE_CLASS_NAME)

DISCOUNT_TERMS Specify whether you want allow discounts, check credit, send dunning
letters, charge interest or and send statements.

CREDIT_CHECKING If you enter a customer profile class in

DUNNING_LETTERS CUSTOMER_PROFILE_CLASS_NAME, your Oracle Receivables
application defaults to the value associated with that profile class if this
column is null.

INTEREST_CHARGES

STATEMENTS If you do not enter a value in CUSTOMER_PROFILE_CLASS_NAME
or you want to override the value associated with the customer profile
class, you must enter a value in these columns.

Validation: Must equal 'Y' (Yes) or 'N' (No)
Mandatory when no profile class specified.

Destination: AR_CUSTOMER_PROFILES.DISCOUNT_TERMS

AR_CUSTOMER_PROFILES.CREDIT_CHECKING

AR_CUSTOMER_PROFILES.
DUNNING_LETTERS

AR_CUSTOMER_PROFILES.
INTEREST_CHARGES

AR_CUSTOMER_PROFILES.STATEMENTS

DISCOUNT_
GRACE_DAYS Enter the number of days after the discount date that this customer can
still take discounts. Do not enter a value in this column if
DISCOUNT_TERMS is either null or set to 'N' (No).

If you enter a customer profile class in
CUSTOMER_PROFILE_CLASS_NAME, your Oracle Receivables
application defaults to the value associated with that profile class if this
column is null.

If you do not enter a value in CUSTOMER_PROFILE_CLASS_NAME or you want to override the value associated with the customer profile class, you must enter a value in this column.

Validation: Number must be non-negative (must be >=0)
Must be null when DISCOUNT_TERMS is null or 'No'

Destination: AR_CUSTOMER_PROFILES.DISCOUNT_
GRACE_DAYS

DUNNING_LETTER_ SET_NAME

Enter the dunning letter set that you want to associate with this customer. You must enter a value if DUNNING_LETTERS is set to 'Y' (Yes). Do not enter a value if DUNNING_LETTERS is either null or set to 'N' (No). Use dunning letters that you previously defined in the Define Dunning Letter Sets form.

If you enter a customer profile class in CUSTOMER_PROFILE_CLASS_NAME, your Oracle Receivables application defaults to the value associated with that profile class if this column is null.

If you do not enter a value in CUSTOMER_PROFILE_CLASS_NAME or you want to override the value associated with the customer profile class, you must enter a value in this column.

Validation: Must exist in AR_DUNNING_LETTER_SETS
Mandatory when DUNNING_LETTERS is Yes
Must be null when DUNNING_LETTERS is No or null

Destination: AR_CUSTOMER_PROFILES.DUNNING_
LETTER_SET_ID
(Derived from DUNNING_LETTER_SET_NAME)

GROUPING_ RULE_NAME

Enter the grouping rule that you want to assign to this customer. Grouping rules are used by AutoInvoice to determine how to create your transactions. Use grouping rules you previously defined in the Define Grouping Rules form.

If you enter a customer profile class in CUSTOMER_PROFILE_CLASS_NAME, your Oracle Receivables application defaults to the value associated with that profile class if this column is null.

If you do not enter a value in CUSTOMER_PROFILE_CLASS_NAME or you want to override the value associated with the customer profile class, you must enter a value in this column.

Validation: Must exist in RA_GROUPING_RULES
Mandatory when no profile class is specified

Destination: AR_CUSTOMER_PROFILES.GROUPING_
RULE_ID (Derived from
GROUPING_RULE_NAME)

INTERFACE_STATUS This column is used by Customer Interface and should be left null. The Customer Interface program updates this column with all error messages which apply to this interface record. If an interface record has several problems, the Customer Interface program updates this column with multiple error codes which are described later in this section.

Validation: None

Destination: None

**INSERT_
UPDATE_FLAG** Enter a value to indicate whether you are inserting a new record or updating an existing record.

Regardless of the value you enter in this column you cannot insert or update profile information in the following columns:
CURRENCY_CODE, AUTO_REC_MIN_RECEIPT_AMOUNT,
INTEREST_RATE, MAX_INTEREST_CHARGE,
MIN_DUNNING_AMOUNT, MIN_DUNNING_INVOICE_AMOUNT,
MIN_FC_BALANCE_AMOUNT, MIN_FC_INVOICE_AMOUNT,
MIN_STATEMENT_AMOUNT, OVERALL_CREDIT_LIMIT,
TRX_CREDIT_LIMIT, AMOUNT_ATTRIBUTE_CATEGORY and
AMOUNT_ATTRIBUTE1 through 15.

If you are trying to insert new profile amount information, Customer Interface will automatically insert this information even if this column is set to 'U'. For example, if you want to update the tax printing option value for a record that you have already inserted and at the same time enter a new currency code for this customer profile, enter 'U' in this column. Customer Interface will automatically update the tax printing option value and automatically insert the new currency code.

This column is required.

Validation: I' for insert, 'U' for update

Destination: None

**INTEREST_
PERIOD_DAYS**

Enter the number of days to which the interest rate refers. You must enter a value if INTEREST_CHARGES is set to 'Y' (Yes). Do not enter a value if INTEREST_CHARGES is either null or set to 'N' (No).

If you enter a customer profile class in CUSTOMER_PROFILE_CLASS_NAME, your Oracle Receivables application defaults to the value associated with that profile class if this column is null.

If you do not enter a value in CUSTOMER_PROFILE_CLASS_NAME or you want to override the value associated with the customer profile class, you must enter a value in this column.

Validation: Number must be positive
Mandatory when INTEREST_CHARGES is Yes
Must be null when INTEREST_CHARGES is No or null

Destination: AR_CUSTOMER_PROFILES.INTEREST_
PERIOD_DAYS

INTEREST_RATE

Enter the interest rate that you want to charge this customer for this currency. This column is optional.

Validation: None

Destination: AR_CUSTOMER_PROFILE_AMOUNTS.
INTEREST_RATE

**LAST_UPDATED_
BY**

Enter the user id that is updating this row. This column is required.

Validation: None

Destination: None

**LAST_UPDATE_
DATE**

Enter the system date. This column is required.

Validation: None

Destination: None

**LAST_UPDATE_
LOGIN**

Enter the login id. This column is optional

Validation: None

Destination: None

**MAX_INTEREST_
CHARGE**

Enter the maximum amount of interest that you want to charge this customer in this currency for each invoice. This column is optional.

Validation: None

	Destination: AR_CUSTOMER_PROFILE_AMOUNTS. MAX_INTEREST_CHARGE
MIN_DUNNING_ AMOUNT	<p>Enter the minimum amount in this currency that must be past due for this customer before you select these customers for dunning. This column is optional.</p> <p>Validation: None</p> <p>Destination: AR_CUSTOMER_PROFILE_AMOUNTS. MIN_DUNNING_AMOUNT</p>
MIN_DUNNING_ INVOICE_AMOUNT	<p>Enter the minimum invoice amount in this currency that must be past due for this customer before you select these customers for dunning. This column is optional.</p> <p>Validation: None</p> <p>Destination: AR_CUSTOMER_PROFILE_AMOUNTS. MIN_DUNNING_INVOICE_AMOUNT</p>
MIN_FC_BALANCE_ AMOUNT	<p>Enter the minimum customer balance that you require before you charge this customer finance charges for past due items in this currency. This column is optional.</p> <p>Validation: None</p> <p>Destination: AR_CUSTOMER_PROFILE_AMOUNTS. MIN_FC_BALANCE_AMOUNT</p>
MIN_FC_INVOICE_ AMOUNT	<p>Enter the minimum invoice balance that you require before you charge this customer finance charges for past due items in this currency. This column is optional.</p> <p>Validation: None</p> <p>Destination: AR_CUSTOMER_PROFILE_AMOUNTS. MIN_FC_INVOICE_AMOUNT</p>
MIN_STATEMENT_ AMOUNT	<p>Enter the minimum outstanding balance in this currency that this customer must exceed in order for your Oracle Receivables application to generate a statement. This column is optional.</p> <p>Validation: None</p> <p>Destination: AR_CUSTOMER_PROFILE_AMOUNTS. MIN_STATEMENT_AMOUNT</p>
ORG_ID	<p>If you use the Oracle Applications multiple organizations feature, enter the operating unit's org_id. If you do not use the Oracle Applications multiple organizations feature, leave this column null. To insert rows</p>

into open interface tables, use the multiple organization table RA_CUSTOMER_PROFILE_INT_ALL and explicitly populate the org_id column. You can determine an operating unit's org_id by querying the HR_OPERATING_UNITS view:

```
SELECT name, organization_id from
HR_OPERATING_UNITS;
```

When you import rows *from* open interface tables, only rows from your current operating unit are processed.

ORIG_SYSTEM_ CUSTOMER_REF

Enter the value that represents the customer or Bill-To customer for which you are inserting or updating customer profile information. If you are entering a new customer in RA_CUSTOMER_INTERFACE, you must enter a customer level profile for this customer.

Validation: For insert, this customer reference must exist in RA_CUSTOMERS or be successfully validated in RA_CUSTOMERS_INTERFACE_ALL
For update, this customer reference must exist in RA_CUSTOMERS

Destination: AR_CUSTOMER_PROFILES.CUSTOMER_ID
(Derived from ORIG_SYSTEM_CUSTOMER_REF)

ORIG_SYSTEM_ ADDRESS_REF

Enter the value that represents the customer Bill-To address for which you are inserting or updating customer profile information. An active Bill-To business purpose must be associated with this address.

Validation: For insert, this address reference must exist in RA_ADDRESSES or be successfully validated in RA_CUSTOMERS_INTERFACE_ALL
For update, this address reference must exist in RA_ADDRESSES

Destination: AR_CUSTOMER_PROFILES.SITE_USE_ID
(Derived from ORIG_SYSTEM_ADDRESS_REF)

OVERALL_ CREDIT_LIMIT

Enter the total amount of credit that you want to give to this customer in this currency. This column is optional.

Validation: TRX_CREDIT_LIMIT and OVERALL_CREDIT_LIMIT must both be filled in, or both be null.
TRX_CREDIT_LIMIT may not be greater than the OVERALL_CREDIT_LIMIT.

OVERRIDE_TERMS

Specify whether you want to be able to enter payment terms that are different from the payment term you enter in STANDARD_TERM_NAME.

If you enter a customer profile class in CUSTOMER_PROFILE_CLASS_NAME, your Oracle Receivables application defaults to the value associated with that profile class if this column is null.

If you do not enter a value in CUSTOMER_PROFILE_CLASS_NAME or you want to override the value associated with the customer profile class, you must enter a value in this column.

Validation: Must equal 'Y' (Yes) or 'N' (No)
Mandatory when no profile class is specified.

Destination: AR_CUSTOMER_PROFILES.OVERRIDE_TERMS

PAYMENT_ GRACE_DAYS

Enter the number of days you allow this customer's receipt to be overdue before you take action.

Validation: Number must be non-negative (must be ≥ 0)

Destination: AR_CUSTOMER_PROFILES.PAYMENT_
GRACE_DAYS

PERCENT_ COLLECTABLE

Enter the percentage of this customer's account balance that you expect to collect regularly. This column is optional.

Validation: Must be between 0 to 100

Destination: AR_CUSTOMER_PROFILES.PERCENT_
COLLECTABLE

REQUEST_ID

This column is used by Customer Interface, and should be left null.

Validation: None

Destination: None

RISK_CODE

Enter the risk code for this customer. Use risk codes you previously defined in the Maintain QuickCodes form with a QuickCode type of 'Customer credit risk'. This column is optional.

Validation: AR_LOOKUPS.LOOKUP_CODE where
LOOKUP_TYPE = 'RISK_CODE'

Destination: AR_CUSTOMER_PROFILES.RISK_CODE

STANDARD_ TERM_NAME

Enter the standard payment terms for this customer. Use payment terms that you previously defined in the Define Payment Terms form. This column is optional.

If you enter a customer profile class in CUSTOMER_PROFILE_CLASS_NAME, your Oracle Receivables

application defaults to the value associated with that profile class if this column is null.

If you do not enter a value in CUSTOMER_PROFILE_CLASS_NAME or you want to override the value associated with the customer profile class, you must enter a value in this column.

Validation: Must exist in RA_TERMS
Must have a unique value

Destination: AR_CUSTOMER_PROFILES.STANDARD_
TERMS (Derived from
STANDARD_TERM_NAME)

STATEMENT_ CYCLE_NAME

Enter the statement cycle that you want to associate to this customer. Use statement cycles that you previously defined in the Define Statement Cycles form. You must enter a value if STATEMENTS is set to 'Y' (Yes). Do not enter a value in this field if STATEMENTS is null or set to 'N' (No).

If you enter a customer profile class in CUSTOMER_PROFILE_CLASS_NAME, your Oracle Receivables application defaults to the value associated with that profile class if this column is null.

If you do not enter a value in CUSTOMER_PROFILE_CLASS_NAME or you want to override the value associated with the customer profile class, you must enter a value in this column.

Validation: Must exist in AR_STATEMENT_CYCLES
Must be null when STATEMENTS is No or null
Mandatory when STATEMENTS is Yes
Must have a unique value

Destination: AR_CUSTOMER_PROFILES.STATEMENT_
CYCLE_ID (Derived from
STATEMENT_CYCLE_NAME)

TAX_PRINTING_ OPTION

Enter either 'Itemize and Summarize', 'Itemize By Line', 'Summarize By Tax Code', or 'Total Tax Only' to indicate how you want to print tax information for this customer.

If you enter a customer profile class in CUSTOMER_PROFILE_CLASS_NAME, your Oracle Receivables application defaults to the value associated with that profile class if this column is null.

If you do not enter a value in CUSTOMER_PROFILE_CLASS_NAME or you want to override the value associated with the customer profile class, you must enter a value in this column.

Validation: AR_LOOKUPS.LOOKUP_CODE where
LOOKUP_TYPE = 'TAX_PRINTING_OPTION'
Mandatory when no profile class specified.

Destination: AR_CUSTOMER_PROFILES.
TAX_PRINTING_OPTION

TOLERANCE

Enter the percent over the credit limit that this customer can exceed before you act.

If you enter a customer profile class in CUSTOMER_PROFILE_CLASS_NAME, your Oracle Receivables application defaults to the value associated with that profile class if this column is null.

If you do not enter a value in CUSTOMER_PROFILE_CLASS_NAME or you want to override the value associated with the customer profile class, you must enter a value in this column.

Validation: Must be between -100 and 100
Mandatory when no profile class specified.

Destination: RA_CUSTOMER_PROFILES.TOLERANCE

TRX_CREDIT_LIMIT

Enter the amount of credit for each order that you want to give to this customer in this currency. This column is optional.

Validation: TRX_CREDIT_LIMIT and OVERALL_CREDIT_LIMIT must both be filled in, or both be null.
TRX_CREDIT_LIMIT may not be greater than the OVERALL_CREDIT_LIMIT.

Destination: AR_CUSTOMER_PROFILE_AMOUNTS.
TRX_CREDIT_LIMIT

VALIDATED_FLAG

This column is used by Customer Interface, and should be left null.

Validation: None

Destination: None

Destination: AR_CUSTOMER_PROFILE_AMOUNTS.
OVERALL_CREDIT_LIMIT

RA_CONTACT_PHONES_INT_ALL

This table stores telephone numbers for customers, addresses and contacts as well as contacts for customers and addresses. You do not have to enter values in this table if you do not want to pass telephone or contact information.

Column Name	Value
CONTACT_ATTRIBUTE_CATEGORY	Enter Descriptive Flexfield category information. These columns are optional. Validation: None Destination: RA_CONTACTS.ATTRIBUTE_CATEGORY RA_PHONES.ATTRIBUTE_CATEGORY
CONTACT_ATTRIBUTE 1 through 25	Enter Descriptive Flexfield information. These columns are optional. Validation: None Destination: RA_CONTACTS.ATTRIBUTE1 through 25
PHONE_ATTRIBUTE 1 through 15	
CONTACT_FIRST_NAME	Enter the contact's first name. This column is optional. Validation: None. Destination: RA_CONTACTS.FIRST_NAME
CONTACT_JOB_TITLE	Enter the job title or responsibility for this contact. Use contact job titles that you previously defined in the Maintain QuickCodes form with a QuickCode type of 'Responsibilities' for the primary customer contact. This column is optional. Validation: None. Destination: RA_CONTACTS.JOB_TITLE.
CONTACT_LAST_NAME	Enter the contact's last name. If ORIG_SYSTEM_CONTACT_REF is filled in, then you must enter a value in this column. Otherwise, this column is optional. Validation: None. Destination: RA_CONTACTS.LAST_NAME

CONTACT_TITLE	Enter the title for this contact. You can enter Mr., Mrs. or Ms. This column is optional. Validation: AR_LOOKUPS.LOOKUP_CODE where LOOKUP_TYPE = 'CONTACT_TITLE' Destination: RA_CONTACTS.TITLE
CREATED_BY	Enter the user id that is creating this row. This column is required. Validation: None Destination: None
CREATION_DATE	Enter the system date. This column is required. Validation: None Destination: None
INTERFACE_STATUS	This column is used by Customer Interface and should be left null. The Customer Interface program updates this column with all error messages which apply to this interface record. If an interface record has several problems, the Customer Interface program updates this column with multiple error codes.
INSERT_UPDATE_FLAG	Enter a value to indicate whether you are inserting a new record or updating an existing record. This column is required. Validation: 'I' for insert, 'U' for update Destination: None
LAST_UPDATED_BY	Enter the user id that is updating this row. This column is required. Validation: None Destination: RA_PHONES.ATTRIBUTE1 through 15
LAST_UPDATE_DATE	Enter the system date. This column is required. Validation: None Destination: None
LAST_UPDATE_LOGIN	Enter the login id. This column is optional Validation: None Destination: None
ORG_ID	If you use the Oracle Applications multiple organizations feature, enter the operating unit's org_id. If you do not use the Oracle Applications

multiple organizations feature, leave this column null. To insert rows into open interface tables, use the multiple organization table RA_CONTACT_PHONES_INT_ALL and explicitly populate the org_id column. You can determine an operating unit's org_id by querying the HR_OPERATING_UNITS view:

```
SELECT name, organization_id from
HR_OPERATING_UNITS;
```

When you import rows *from* open interface tables, only rows from your current operating unit are processed.

ORIG_SYSTEM_ CUSTOMER_REF

Enter a value that you can use to uniquely identify this customer in your original system. The reference number you enter here provides you with an audit trail from your Oracle Receivables application back to your original system. If you are entering either contact or telephone information, then you must enter a value in this column.

Validation: Must equal RA_CUSTOMERS.ORIG_SYSTEM_REFERENCE for update

Destination: RA_CUSTOMERS.ORIG_SYSTEM_REFERENCE

ORIG_SYSTEM_ ADDRESS_REF

Enter a value that you can use to uniquely identify this address in your original system. The reference number that you enter here provides you with an audit trail from your Oracle Receivables application to your original system.

If you are entering information that refers to an address, such as a contact or telephone, then you must enter a value in this column. Otherwise, this column is optional.

If this column is null, then the phone or contact that you enter refers to the customer.

Validation: Must equal RA_ADDRESSES.ORIG_SYSTEM_REFERENCE for update

Destination: RA_ADDRESSES.ORIG_SYSTEM_REFERENCE

ORIG_SYSTEM_ CONTACT_REF

Enter a value that you can use to uniquely identify this contact in your original system. The reference number that you enter here provides you with an audit trail from your Oracle Receivables application back to your original system.

This column forms part of the primary key for RA_CONTACT_PHONES_INTERFACE. The primary key is a

combination of ORIG_SYSTEM_CONTACT_REF and ORIG_SYSTEM_TELEPHONE_REF.

If you are entering contact information, or information that refers to a contact such as a telephone number assigned to a contact, then you must enter a value in this column. Otherwise, this column is optional.

To enter a contact for a customer, do not enter a value in ORIG_SYSTEM_ADDRESS_REF.

If you are entering a contact for a specific address, then enter values in both ORIG_SYSTEM_CUSTOMER_REF and ORIG_SYSTEM_ADDRESS_REF.

Validation: Must equal RA_CONTACTS.ORIG_SYSTEM_REFERENCE for update

Destination: RA_CONTACTS.ORIG_SYSTEM_REFERENCE



Suggestion: If the value you enter is numeric, add an alpha character to the end to ensure that this number never conflicts with a system-generated ID number.

ORIG_SYSTEM_ TELEPHONE_REF

Enter a value that you can use to uniquely identify this telephone in your original system. The reference value that you enter here provides you with an audit trail from your Oracle Receivables application back to your original system.

This column forms part of the primary key for RA_CONTACT_PHONES_INTERFACE. The primary key is a combination of ORIG_SYSTEM_CONTACT_REF and ORIG_SYSTEM_TELEPHONE_REF.

If you are entering telephone information, then you must enter this column. Otherwise, this column is optional.

To enter a telephone for a customer, do not enter values in ORIG_SYSTEM_ADDRESS_REF or ORIG_SYSTEM_CONTACT_REF.

To enter a telephone for a specific address, enter values in ORIG_SYSTEM_CUSTOMER_REF and ORIG_SYSTEM_ADDRESS_REF.

To enter telephones for a specific contact, enter values in ORIG_SYSTEM_CUSTOMER_REF, ORIG_SYSTEM_CONTACT_REF, and ORIG_SYSTEM_ADDRESS_REF, if the contact is associated with an address.

Validation: Must equal RA_PHONES.ORIG_SYSTEM_REFERENCE
Mandatory when specifying telephone information

Destination: RA_PHONES.ORIG_SYSTEM_REFERENCE



Suggestion: If the value you enter is numeric, add an alpha character to the end to ensure that this number never conflicts with a system-generated ID number.

REQUEST_ID

This column is used by Customer Interface, and should be left null.

TELEPHONE

Enter the telephone number for the customer, address, or contact. If ORIG_SYSTEM_TELEPHONE_REF is filled in, then you must enter a value for this column. Otherwise, leave this column null.

Validation: Mandatory when specifying telephone information.

Destination: RA_PHONES.PHONE_NUMBER

TELEPHONE_ AREA_CODE

Enter the area code or extension for the telephone number, depending on the column you choose. These columns are optional.

Validation: None

TELEPHONE_ EXTENSION

Destination: RA_PHONES.AREA_CODE

RA_PHONES.EXTENSION

TELEPHONE_TYPE

Enter the type of telephone number such as General, Fax, or Telex. Use telephone types that you previously defined in the Maintain QuickCodes form with a QuickCode type of 'Types of communication used in contacting customers'.

If ORIG_SYSTEM_TELEPHONE_REF is filled in, then you must enter a value in this column. Otherwise, leave this column null.

Validation: AR_LOOKUPS.LOOKUP_CODE where
LOOKUP_TYPE = 'COMMUNICATION_TYPE'
Mandatory when specifying telephone information

Destination: RA_CONTACTS.PHONE_TYPE

VALIDATED_FLAG

This column is used by Customer Interface, and should be left null.

RA_CUSTOMER_BANKS_INT_ALL

This table stores bank information for a customer or for a specific Bill-To address. You do not have to enter values in this table if you do

not want to pass bank information. Your Oracle Receivables application uses customer bank information when creating automatic receipts such as Bills of Exchange and Direct Debits. If you associate an automatic payment method to a customer or a customer's Bill-To business purpose, you must enter a bank account for this customer.

Column Name	Value
ATTRIBUTE_CATEGORY	<p>Enter Descriptive Flexfield category information. This column is optional.</p> <p>Validation: None</p> <p>Destination: AP_BANK_ACCOUNT_USES.ATTRIBUTE_CATEGORY</p>
ATTRIBUTE 1 through 15	<p>Enter Descriptive Flexfield information. This column is optional.</p> <p>Validation: None</p> <p>Destination: AP_BANK_ACCOUNT_USES.ATTRIBUTE 1 through 15</p>
BANK_ACCOUNT_NUM	<p>Enter the account number or currency code for this bank account, depending on the column you choose. If the bank account already exists, do not enter a value. If the bank account does not exist, you must enter a value.</p> <p>Validation: Mandatory when the bank account is not defined.</p> <p>Destination: AP_BANK_ACCOUNT.BANK_ACCOUNT_NUM</p>
BANK_ACCOUNT_CURRENCY_CODE	<p>Destination: AP_BANK_ACCOUNTS.CURRENCY_CODE</p>
BANK_ACCOUNT_INACTIVE_DATE	<p>Enter the date that this bank account becomes inactive. This column is optional.</p> <p>Validation: None</p> <p>Destination: AP_BANK_ACCOUNTS.INACTIVE_DATE</p>
BANK_ACCOUNT_DESCRIPTION	<p>Enter a description for this bank account. This column is optional.</p> <p>Validation: None</p> <p>Destination: AP_BANK_ACCOUNTS.DESCRPTION</p>
BANK_ACCOUNT_CHECK_DIGITS	<p>Enter the number this bank account prints on checks. This column is optional.</p>

	Validation: None Destination: AP_BANK_ACCOUNT.CHECK_DIGITS
BANK_ACCOUNT_NAME	<p>Enter the bank account name you want to assign to this customer or Bill-To address. Use the Define Customer Banks form to define banks and bank accounts for your customers. If the bank account has not already been defined, Customer Interface will try to create it in AP_BANK_ACCOUNTS. This column is required only if you are inserting bank information.</p> Validation: Must exist in AP_BANK_ACCOUNTS or if it doesn't exist, values must also exist for BANK_ACCOUNT_CURRENCY_CODE, BANK_ACCOUNT_NUM, BANK_NAME and BANK_BRANCH_NAME Destination: AP_BANK_ACCOUNTS.BANK_ACCOUNT_NAME
BANK_NAME BANK_BRANCH_NAME	<p>Enter the name of the bank or bank branch for the account you are inserting. If the bank account already exists, do not enter a value. If the bank account does not exist, you must enter a value.</p> Validation: BANK_NAME together with BANK_BRANCH_NAME must be unique. Mandatory when the bank account is not defined. Destination: AP_BANK_ACCOUNTS.BANK_NAME AP_BANK_BRANCHES.BANK_BRANCH_NAME
BANK_NUMBER	<p>Enter the number of the bank associated with the bank account you are inserting. If the bank account already exists, do not enter a value. If the bank account does not exist, you can enter a value.</p> Validation: Must be unique Destination: AP_BANK_BRANCHES.BANK_NUMBER
BANK_NUM	<p>Enter the number of the bank branch associated with the bank account you are inserting. If the bank account already exists, do not enter a value. If the bank account does not exist, you can enter a value.</p> Validation: Must be unique. Destination: AP_BANK_BRANCHES.BANK_NUM
BANK_BRANCH_DESCRIPTION	<p>Enter a description for this bank branch. This column is optional.</p>

	Validation: None Destination: AP_BANK_BRANCHES.DESCRPTION
BANK_BRANCH_ ADDRESS 1 through 4	Enter the street address, city, county, state, postal code, or province for this bank branch. These columns are optional.
BANK_ BRANCH_CITY	
BANK_ BRANCH_COUNTRY	
BANK_ BRANCH_STATE	
BANK_ BRANCH_ZIP	
BANK_ BRANCH_PROVINCE	
	Validation: None Destination: AP_BANK_BRANCHES.ADDRESS_LINE1-3 AP_BANK_BRANCHES.CITY AP_BANK_BRANCHES.STATE AP_BANK_BRANCHES.ZIP AP_BANK_BRANCHES.PROVINCE
BANK_ BRANCH_COUNTRY	Enter the country for this bank branch. This column is optional.
	Validation: Must exist in FND_TERRITORIES.TERRITORY_CODE Destination: AP_BANK_BRANCHES.COUNTRY
BANK_ BRANCH_PHONE	Enter the telephone number or telephone area code for this bank branch. These columns are optional.

**BANK_
BRANCH_
AREA_CODE**

Validation: None
Destination: AP_BANK_BRANCHES.PHONE
AP_BANK_BRANCHES.AREA_CODE

**BANK_BRANCH_EFT
_USER_NUMBER** Enter the Electronic Funds Transfer user id. This column is optional.

Validation: None
Destination: AP_BANK_BRANCHES.EFT_USER_NUMBER

**BANK_ACCOUNT_
ATT_CATEGORY** Enter Descriptive Flexfield category information. These columns are optional.

**BANK_
BRANCH_
ATT_CATEGORY**

Validation: None
Destination: AP_BANK_ACCOUNTS.ATTRIBUTE_
CATEGORY
AP_BANK_BRANCHES.ATTRIBUTE_CATEGORY

**BANK_ACCOUNT_
ATTRIBUTE
1 through 15** Enter Descriptive Flexfield information. These columns are optional.

**BANK_
BRANCH_
ATTRIBUTE
1 through 15**

Validation: None
Destination: AP_BANK_ACCOUNTS.ATTRIBUTE1 through 15
AP_BANK_BRANCHES.ATTRIBUTE1 through 15

CREATED_BY Enter the user id that is creating this row. This column is required.

Validation: None
Destination: None

CREATION_DATE Enter the system date. This column is required.

Validation: None

	Destination: None
END_DATE	Enter the date that this bank account becomes inactive. This column is optional.
	Validation: End Date cannot be before the start date. Customers can be assigned to multiple bank accounts as long as there is no overlapping date range.
	Destination: AP_BANK_ACCOUNT_USES.END_DATE
INTERFACE_STATUS	This column is used by Customer Interface and should be left null. The Customer Interface program updates this column with all error messages which apply to this interface record. If an interface record has several problems, the Customer Interface program updates this column with multiple error codes.
	Validation: None
	Destination: None
LAST_UPDATED_BY	Enter the user id that is updating this row. This column is required.
	Validation: None
	Destination: None
LAST_UPDATE_DATE	Enter the system date. This column is required.
	Validation: None
	Destination: None
LAST_UPDATE_LOGIN	Enter the login id. This column is optional
	Validation: None
	Destination: None
ORG_ID	<p>If you use the Oracle Applications multiple organizations feature, enter the operating unit's org_id. If you do not use the Oracle Applications multiple organizations feature, leave this column null. To insert rows into open interface tables, use the multiple organization table RA_CUSTOMER_BANKS_INT_ALL and explicitly populate the org_id column. You can determine an operating unit's org_id by querying the HR_OPERATING_UNITS view:</p> <pre>SELECT name, organization_id from HR_OPERATING_UNITS;</pre>

When you import rows *from* open interface tables, only rows from your current operating unit are processed.

**ORIG_SYSTEM_
CUSTOMER_REF**

Enter the value that represents the customer for which you are inserting bank information. This column is required only if you want to insert bank information.

Validation: For insert, the customer reference must exist in RA_CUSTOMERS or be successfully validated in RA_CUSTOMERS_INTERFACE_ALL

Destination: AP_BANK_ACCOUNT_USES.CUSTOMER_ID
(Derived from ORIG_SYSTEM_CUSTOMER_REF)

**ORIG_SYSTEM_
ADDRESS_REF**

Enter the value that represent the customer address for which you are inserting bank information. An active Bill-To business purpose must be associated with this address. This column is required only if you want to insert bank information for a specific Bill-To address.

Validation: For insert, the address reference must exist in RA_ADDRESSES or be successfully validated in RA_CUSTOMERS_INTERFACE_ALL
(Derived from ORIG_SYSTEM_CUSTOMER_REF)

Destination: AP_BANK_ACCOUNT_USES.CUSTOMER_SITE_USE_ID

PRIMARY_FLAG

Enter 'Y' or 'N' to indicate whether this is the primary bank account for this customer or Bill-To address. This column is required only if you are inserting bank information.

Validation: Only one primary bank account can exist at either the customer level or address level

Destination: AP_BANK_ACCOUNT_USES.PRIMARY_FLAG

REQUEST_ID

This column is used by Customer Interface, and should be left null.

Validation: None

Destination: None

START_DATE

Enter the starting date that this bank account becomes active. This column is required only if you are inserting bank information.

Validation: End Date cannot be before the start date. Customers can be assigned to multiple bank accounts as long as there is no overlapping date range.

Destination: AP_BANK_ACCOUNT_USES.START_DATE

VALIDATED_FLAG This column is used by Customer Interface, and should be left null.

Validation: None

Destination: None

RA_CUST_PAY_METHOD_INTERFACE

This table stores payment method information for a customer or for a specific Bill-To address. You do not have to enter values in this table if you do not want to pass a payment method. However, if you associate an automatic payment method to a customer or a customer's Bill-To address, a bank account must exist for this customer. Your Oracle Receivables application uses automatic payment methods when creating automatic receipts such as Bills of Exchange and Direct Debits.

Column Name

Value

ATTRIBUTE_ CATEGORY

Enter Descriptive Flexfield category information. This column is optional.

Validation: None

Destination: RA_CUST_RECEIPT_METHODS.
ATTRIBUTE_CATEGORY

ATTRIBUTE 1 through 15

Enter Descriptive Flexfield information. This column is optional.

Validation: None

Destination: RA_CUST_RECEIPT_METHODS.ATTRIBUTE
1 through 15

CREATED_BY

Enter the user id that is creating this row. This column is required.

Validation: None

Destination: None

CREATION_DATE

Enter the system date. This column is required.

Validation: None

Destination: None

END_DATE

Enter the date that this payment method becomes inactive. This column is optional.

Validation: Customers can be assigned to multiple payment methods as long as there is no overlapping date range.

Destination: RA_CUST_RECEIPT_METHODS.END_DATE

INTERFACE_STATUS This column is used by Customer Interface and should be left null. The Customer Interface program updates this column with all error messages which apply to this interface record. If an interface record has several problems, the Customer Interface program updates this column with multiple error codes.

Validation: None

Destination: None

LAST_UPDATED_BY Enter the user id that is updating this row. This column is required.

Validation: None

Destination: None

LAST_UPDATE_DATE Enter the system date. This column is required.

Validation: None

Destination: None

LAST_UPDATE_LOGIN Enter the login id. This column is optional

Validation: None

Destination: None

ORG_ID If you use the Oracle Applications multiple organizations feature, enter the operating unit's org_id. If you do not use the Oracle Applications multiple organizations feature, leave this column null. To insert rows into open interface tables, use the multiple organization table RA_CUST_PAY_METHOD_INT_ALL and explicitly populate the org_id column. You can determine an operating unit's org_id by querying the HR_OPERATING_UNITS view:

```
SELECT name, organization_id from
HR_OPERATING_UNITS;
```

When you import rows *from* open interface tables, only rows from your current operating unit are processed.

ORIG_SYSTEM_CUSTOMER_REF Enter the value that represents the customer for which you are inserting a payment method. This column is required only if you want to insert a payment method.

	<p>Validation: For insert, the customer reference must exist in RA_CUSTOMERS or be successfully validated in RA_CUSTOMERS_INTERFACE_ALL</p> <p>Destination: RA_CUST_RECEIPTS_METHODS. CUSTOMER_ID (Derived from ORIG_SYSTEM_CUSTOMER_REF)</p>
ORIG_SYSTEM_ADDRESS_REF	<p>Enter the value that represents the customer address for which you are inserting a payment method. An active Bill-To business purpose must be associated with this address. This column is required only if you are inserting a payment method for a specific Bill-To address.</p> <p>Validation: For insert, this address reference must exist in RA_ADDRESSES or be successfully validated in RA_CUSTOMERS_INTERFACE_ALL</p> <p>Destination: RA_CUST_RECEIPT_METHODS.SITE_USE_ID (Derived from ORIG_SYSTEM_ADDRESS_REF)</p>
PAYMENT_METHOD_NAME	<p>Enter the name of the payment method that you want to assign to this customer or Bill-To address. This column is only required if you are inserting a payment method.</p> <p>Validation: Must exist in AR_RECEIPT_METHODS</p> <p>Destination: RA_CUST_RECEIPTS_METHODS. RECEIPT_METHOD_ID (Derived from PAYMENT_METHOD_NAME)</p>
PRIMARY_FLAG	<p>Enter 'Y' or 'N' to indicate whether this is the primary payment method for this customer or Bill-To address. This column is required only if you are inserting a payment method.</p> <p>Validation: Only one primary payment method can exist at either the customer level or Bill-To address level.</p> <p>Destination: RA_CUST_RECEIPTS_METHODS. PRIMARY_FLAG</p>
REQUEST_ID	<p>This column is used by Customer Interface, and should be left null.</p> <p>Validation: None</p> <p>Destination: None</p>
START_DATE	<p>Enter the starting date that this payment method becomes active. This column is required only if you are inserting payment method information.</p>

	Validation:	Customers can be assigned to multiple payment methods as long as there is no overlapping date range.
	Destination:	RA_CUST_RECEIPT_METHODS.ST ART_DATE
VALIDATED_FLAG		This column is used by Customer Interface, and should be left null.
	Validation:	None
	Destination:	None

Valid Relationships

Customers

A customer may have many addresses.

A customer may have many contacts.

A customer may have many telephone numbers.

A customer may be associated with other customers.

A customer may have one primary address for each business purpose such as BILL_TO, SHIP_TO, and so on.

Addresses

An address may have many business purposes.

An address may have many contacts.

An address may have many telephone numbers.

Contacts

A contact may have many telephone numbers.

How to Achieve Desired Results

Customers With Multiple Addresses

To enter a customer with two addresses, you must enter two records in RA_CUSTOMERS_INTERFACE_ALL with identical

ORIG_SYSTEM_CUSTOMER_REF values, but different
ORIG_SYSTEM_ADDRESS_REF values.

Customers With Multiple Contacts

To enter a customer with two contacts, you must enter two records in RA_CONTACT_PHONES_INT_ALL with identical
ORIG_SYSTEM_CUSTOMER_REF values, but different
ORIG_SYSTEM_CONTACT_REF values.

Customers With Multiple Telephone Numbers

To enter a customer with two telephone numbers, you must enter two records in RA_CONTACT_PHONES_INT_ALL with identical
ORIG_SYSTEM_CUSTOMER_REF values, but different
ORIG_SYSTEM_TELEPHONE_REF values.

Addresses With Multiple Business Purposes

To enter an address with two business purposes, you must enter two records in RA_CUSTOMERS_INTERFACE_ALL with identical
ORIG_SYSTEM_CUSTOMER_REF and
ORIG_SYSTEM_ADDRESS_REF values, but different
SITE_USE_CODES values.

Addresses With Multiple Contacts

To enter an address with two contacts, you must enter two records in RA_CONTACT_PHONES_INT_ALL with identical
ORIG_SYSTEM_CUSTOMER_REF and
ORIG_SYSTEM_ADDRESS_REF values, but different
ORIG_SYSTEM_CONTACT_REF values.

Addresses With Multiple Telephone Numbers

To enter an address with two telephone numbers, you must enter two records in RA_CONTACT_PHONES_INT_ALL with identical
ORIG_SYSTEM_CUSTOMER_REF and
ORIG_SYSTEM_ADDRESS_REF values, but different
ORIG_SYSTEM_TELEPHONE_REF values.

Contacts With Multiple Telephone Numbers

To enter a contact with two telephone numbers, you must enter two records in RA_CONTACT_PHONES_INT_ALL with identical

ORIG_SYSTEM_CUSTOMER_REF and
ORIG_SYSTEM_CONTACT_REF values, but different
ORIG_SYSTEM_TELEPHONE_REF values.

Generating Original System References

How do you choose useful ORIG_SYSTEM_%_REF values?

Choose a value that you can easily derive from your original customer database. For example, if your original customer database has a 4 digit customer id, a 6 digit address id, and a 5 digit contact id, and a 7 digit telephone id, you could create a standard format for these values such as:

ORIG_SYSTEM_CUSTOMER_REF

CUST0001

ORIG_SYSTEM_ADDRESS_REF

CUST0001-ADDR000001

ORIG_SYSTEM_CONTACT_REF

CUST0001-ADDR000001-CONT00001

ORIG_SYSTEM_TELEPHONE_REF

CUST0001-ADDR000001-CONT00001-TELE0000001

Integrating Oracle Receivables Applications Tax Information Using Sales Tax Rate Interface

The Sales Tax Rate Interface enables you to load sales tax records into your application from your sales tax feeder system. You can use the Sales Tax Rate Interface program to load new locations and tax rates from an interface area into your Oracle Receivables application. This program allows you to load one or more tax rates for the same location which cover different date ranges and postal codes. The records that you load into your Oracle Receivables application through the Sales Tax Rate Interface program update your existing sales tax rates with the most current tax rates for each location.

Basic Needs

Oracle Receivables and Oracle Government Receivables provide you with the following features you need to load the most current tax data into your receivables system. You should be able to:

- Define new locations for your tax authorities
- Define new locations for address validation
- Update existing sales tax rates
- Add new sales tax rates
- Terminate existing sales tax rates
- Assign multiple sales tax rates to the same location that have different date and postal code ranges

Major Features

Validate Addresses

Use the locations that you transfer into your Oracle Receivables application through the Sales Tax Rate Interface program to validate new customer addresses. The value you enter in the Address Validation field of the QuickCash form determines the type of address validation you want to do. Your Oracle Receivables application only validates customer addresses for the country you specify for the default country field in your system options.

Alternatively, you can implement flexible address formats to provide country specific address entry and validation rules.



Using Flexible Address Formats
(*Oracle Receivables Reference Manual*)

Adjust Tax Rate Dates

The Sales Tax Rate Interface program can adjust the active date range of existing tax rates as new tax rates are loaded into your Oracle Receivables application.

Load Multiple Tax Locations and Rates

The Sales Tax Rate Interface program allows you to load all of your locations and tax rates into your Oracle Receivables application at one time rather than enter each one manually. You can also load rates for sales tax rate overrides. You can immediately use this new tax information to create invoices manually, import invoices into your Oracle Receivables application using the AutoInvoice program, and validate new customer addresses that you enter.

Maintain Up to Date Tax Rate Information

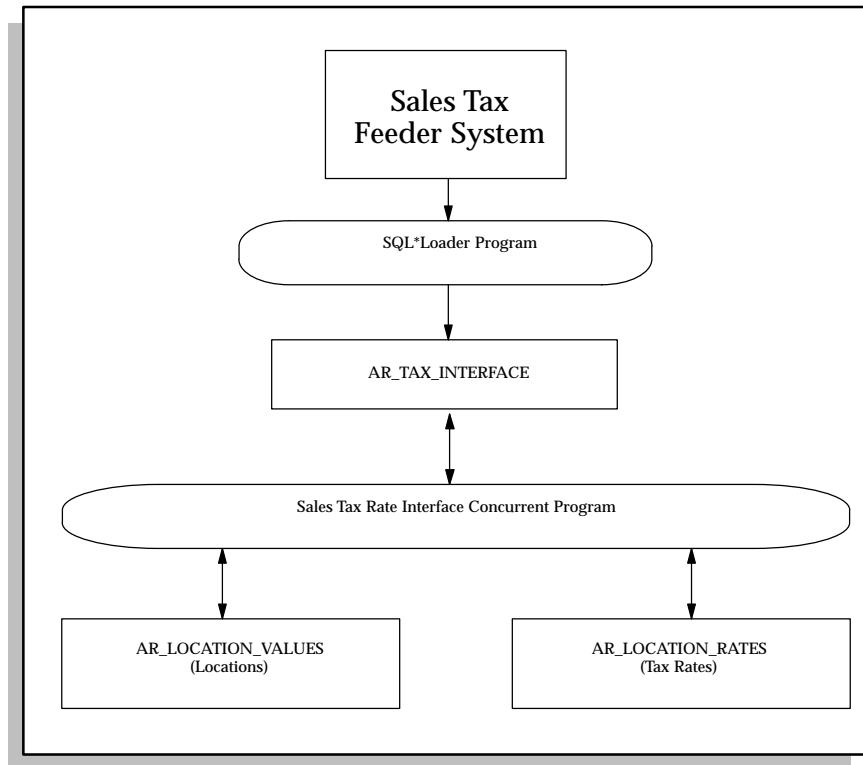
Use the Sales Tax Rate Interface program to ensure that you have the most current sales tax information in your Oracle Receivables application. You can submit this program periodically to update your existing tax rates with the most recent changes.

Overview

The following diagram shows how customer information is imported into the your Oracle Receivables application tax tables.

You should only use this program to load locations and tax rates for the country you specify for you system options in the Default Country field.

Figure 10 – 8



Refer to your Implementation Manual to see the validation that Sales Tax Rate Interface performs on each column of the AR_TAX_INTERFACE table.

Preparing to Load Sales Tax Rates

To ensure that the Sales Tax Rate Interface works correctly, you should complete the following actions before loading sales tax records into your Oracle Receivables application.

You only have to perform the following preparation step for the Sales Tax Rate Interface once.

Verify the Sales Tax Location Flexfield Structure

When you use the Tax Interface program for the first time, navigate to the Define Key Flexfields form and verify that your Sales Tax Location Flexfield structure is frozen and compatible with the location and tax

rate data that you will be loading into your Oracle Receivables application.

You must perform the following preparation steps for the Sales Tax Rate Interface program every time you execute the Sales Tax Rate Interface program.

Perform Backup of Sales Tax Information

Before you load sales tax data into your Oracle Receivables application, you should perform a backup of your location and tax rate information. This information is stored in the following tables:

- . AR_LOCATION_VALUES
- . AR_LOCATION_RATES
- . AR_LOCATION_COMBINATIONS
- . AR_SALES_TAX

Clear Tax Interface Table

The SQL Loader control scripts aravpctl and arvertexctl will clear all existing tax rate information prior to loading a new set of locations and rates. You can either rely upon these SQL Loader scripts to clear existing records or truncate the table AR_TAX_INTERFACE manually.

Verify and Define Tax Table Storage Parameters

Verify that the AR_LOCATION_VALUES and AR_LOCATION_RATES tables into which you will load locations and tax rates have sufficient allocations of space. If you are loading sales tax data for the entire United States, then your database administrator must allocate the following amount of space for your Oracle Receivables application tax interface, tax locations, and tax rates tables.

Table	Data Storage	Index Storage
AR_TAX_INTERFACE	8 Megabytes	11 Megabytes
AR_LOCATION_VALUES	7 Megabytes	8 Megabytes
AR_LOCATION_RATES	10 Megabytes	5 Megabytes
TOTAL:	25 Megabytes	24 Megabytes

Table 10 – 21 Tax Storage Parameters (Page 1 of 1)

A submission of sales tax information which includes every state, county, and city in the United States will load approximately 60, 000 records into the AR_TAX_INTERFACE table.

If you enter customer addresses which use every state, county, and city within the United States, your Oracle Receivables application creates sales tax rates to handle each of these distinct addresses. Your Oracle Receivables application uses the table

AR_LOCATION_COMBINATIONS to store each unique authority, and the table AR_SALES_TAX to store the composite sales tax rates that it generates for these authorities. In this case, your database will require the following storage space for these tables in addition to the storage requirements listed previously.

Table	Data Storage	Index Storage
AR_LOCATION_COMBINATIONS	7 Megabytes	6 Megabytes
AR_SALES_TAX	8 Megabytes	5 Megabytes
TOTAL:	15 Megabytes	11 Megabytes

Table 10 – 22 Authority and Sales Tax Rates Storage Parameters (Page 1 of 1)

Importing Data From Your Feeder System

After you finish preparing to run the Sales Tax Rate Interface program, use a SQL*Loader script to transfer sales tax records from your sales tax feeder system into the tax interface table.

Your Oracle Receivables application provides two SQL*Loader scripts **arvertex.ctl** and **aravp.ctl** as model files for loading your sales tax data into the tax interface table from your feeder system. These scripts are designed for the 'State.County.City' Sales Tax Location Flexfield structure and must be modified if you are using a different structure. The scripts reside in your **ar/bin** directory.

Note: These SQL*Loader control files upload current sales tax rate data only. If you need to upload historical sales tax data, you will have to customize the supplied scripts. Additionally, you should check with your Tax Vendor to ensure they can supply you with historic sales tax rates.

The SQL*Loader scripts support several sales tax features. You can use these scripts to define sales tax locations and tax rates for each of the segments in the structure. The scripts support the loading of multiple tax rates for a single tax location that have different effectivity dates, and multiple zip code ranges and handle the use of common names which, for example, are used for cities, and areas within cities. They will also upload rates for sales tax rate overrides.

Sales Tax Rate Interface Validation

By understanding the data in the AR_TAX_INTERFACE table, you can identify the underlying causes of invalid tax interface data. If the Sales Tax Rate Interface program identifies a record in the interface table that has invalid data, then the program will not pass this record into your Oracle Receivables application location and tax rate tables.

Each location that you load into the tax interface table must have a unique location ID associated with it. This unique location ID enables the Sales Tax Rate Interface program to distinguish between different locations even if the same name appears more than once in your tax data. All locations which are junior to other locations must identify their parent through the parent location ID. The Sales Tax Rate Interface program will only load a tax record into your Oracle Receivables application if it has a parent, unless it is a senior location.

In the example below, each location has a unique location ID. This location ID is composed of a unique identifier for the junior location prefixed with the location ID of the parent of this location. To identify their parents, the county of Los Angeles has the location ID of California as its parent location ID, and the city of Los Angeles has the location ID of Los Angeles county as its parent location ID. The Sales Tax program will not load Record 4 into your Oracle Receivables

application because this record does not have a parent location ID, and its qualifier indicates that this is not a senior location.

Record Number	Location ID	Parent Location ID	Location Value	Location Qualifier
1	11		California	State
2	1101	11	Los Angeles	County
3	110102	1101	Los Angeles	City
4	110103		Santa Monica	City
5	1104	11	Orange County	County
4	110405	1104	Santa Ana	City

Table 10 – 23 Location and Parent Location IDs (Page 1 of 1)

If a tax record that you load into the tax interface table from your sales tax feeder system includes a location without rates, then the Sales Tax Rate Interface program will load this location information. You must assign tax rates to this location manually using the Define Tax Locations and Rates form in your Oracle Receivables application.

If the tax record that you load into the interface table has a tax account segment, your Oracle Receivables application will automatically default the tax account defined in the Define Tax Codes and Rates form for the tax code of type 'Location', as the tax account for this segment. The tax account will be populated in the AR_LOCATION_VALUES.TAX_ACCOUNT_CCID.

If no tax code of type 'Location' has been defined, your Oracle Receivables application will default the Tax Location Account defined in the Define System Options form as the tax account for this segment.

Each location that you pass into the tax interface table must have an appropriate segment qualifier value. You assign a segment qualifier to each segment of your Sales Tax Location Flexfield structure. For example, if you are loading in the city of Los Angeles, then this location must have a segment qualifier of 'CITY'. The Sales Tax Rate Interface program will not load tax records into your Oracle Receivables application that do not have a segment qualifier.

Each record in the tax interface table must have a value for the CHANGE_FLAG column. This value indicates to the Sales Tax Rate Interface program whether a record is different from the most recent listing of this tax data. If this column is 'N', then the Sales Tax Rate Interface program assumes this record is unchanged. If it is 'Y', then

the program assumes that this record is different from the most recent listing. You can use this column to only submit the Sales Tax Rate Interface program for tax data which has been changed since the last submission. This could reduce the time it takes to upload your Sales Tax data.

The Sales Tax Rate Interface program assigns Action Codes to each row of the Sales Tax Rate Interface table based on the results of the validation it performed for that row.

Using Sales Tax Rate Interface

Report Submission Parameters

Parameter	Possible Values
Review or Upload	Load All Data in Tax Interface
	Load Changed Data Only in Tax Interface Table
	Review Tax Interface Data
Print Format	Print All Detail
	Print Summary Only
	Print Warnings in Detail
Senior Location	Enter a value from the first segment of your Sales Tax Location Flexfield structure containing the sales tax rates you want to upload or review. For example, if your structure is State.County.City then you would enter the State that contained the rates you want to upload or review. If you leave this parameter null, then all values are included

Table 10 – 24 Sales Tax Rate Interface Submission Parameters (Page 1 of 1)

Review or Upload

If you enter 'Load All Data in Tax Interface' for the Review or Upload parameter, the Sales Tax Rate Interface program loads all tax data from the tax interface table into the location and tax rates tables.

If you enter 'Load Changed Data Only in Tax Interface Table' for this parameter, the program only loads tax rows from the tax interface table which are different from the most recent data your sales tax feeder system supplied.

If you enter 'Review Tax Interface Data' for this parameter, the program prints the Tax Interface report without actually performing the upload to the location and tax rate tables. Use this to review the tax data in the tax interface table before or after you have submitted the Sales Tax Rate Interface program. If you use this method after invoking the Sales Tax Rate Interface program, the Tax Interface report shows the action that it performed on each row of the tax data in the tax interface table.

Choose 'Review Tax Interface Data' if you want to review the tax rates of a particular location in the interface table before loading the records. In this case, enter 'Print All Detail' as your Print Format parameter value, and specify the senior location for which you want to review tax rates. You can then use SQL*Plus to update the tax rates for this location if they are not what you require.

Print Format

If you enter 'Print Summary Only' as your Print Format parameter, the Tax Interface report includes a summary of the actions that the Sales Tax Rate Interface program has performed on the tax data uploaded from the tax interface table. This summary includes each action code that the Sales Tax Rate Interface program used for the upload, the action code's description, and the number of rows on which the program executed each action.

These action codes are described in detail later in the Action Codes section of this essay.

If you enter 'Print Warnings in Detail' as your Print Format parameter, the Tax Interface report includes a summarized section that lists the program's actions, and a detailed section that lists lines from the tax interface table that the program identified with a warning action code.

If you enter 'Print All Detail' as your Print Format parameter, the Tax Interface report includes a summarized section that lists the program's actions, and a detailed section that lists each line that the program loaded and the action that was identified for it.

Senior Location

If you do not enter a value for the senior location parameter, the Sales Tax Rate Interface program allows you to upload or review tax rows in the tax interface table for all senior locations. Use QuickPick to select a

specific location as your senior location value if you want to upload or review only the tax rows for a specific location. If you are using one of the predefined Sales Tax Location Flexfield structures, the first segment of that structure will be your senior location.

Tax Interface Report

Figure 10 – 9

Senior Segment: CA For Request: All			Tax Interface Report				Date: 17-AUG-1993 12:15 Page: 1			
Printing Option:Print Warnings in Detail			STATE: CA - California							
Action: Review Tax Interface Data			Summary for California							
Action Code		Action Description				Number of Rows				
ALREADY-EXISTS		Tax Rate Already Exists				1				
IGNORE		Ignored				1				
INSERT		New Tax Rate for Existing Location				231				
NEW-LOCATION-INSERT		New Location Created, Tax Rate information Defined				2608				
Location Number		Line	Qualifier	Value	Warnings for California From Zip To Zip		Start Date	End Date	Tax Rate	Action
1		178421	CITY	SOUTH LAKE TAHOE	96150 96150				0.00	IGNORE
2		178618	CITY	WHITEHORN	95589 95589				0.00	ALREADY-EXISTS
Locations and Rates Defined for Warnings										
Location Number		Location			From Zip	To Zip	Start Date	End Date	Tax Rate	
1		EL DORADO-SOUTH LAKE TAHOE			95702	95708	01-JAN-1900	31-DEC-2199	0.00	
					95716	95716	01-JAN-1900	31-DEC-2199	0.00	
					95761	95761	01-JAN-1900	31-DEC-2199	0.00	
					96150	96158	01-JAN-1900	31-DEC-2199	0.00	
2		HUMBOLDT-WHITEHORN			95495	95495	01-JAN-1900	31-DEC-2199	0.00	
					95589	95589	01-JAN-1900	31-DEC-2199	0.00	
Orphan Records, Parent_Location_ID was invalid										
Line		Qualifier	Location Value		From Zip	To Zip	End Date	Start Date	Tax Rate	Location Id Parent ID
176232		CITY	BORDER		99780	99780			0.00	022401056 02240
176233		CITY	SLANA		99586	99586			0.00	022611037 02261

Review Sales Tax Rate Interface Report

The Tax Interface report, which is generated during execution of the Sales Tax Rate Interface program, provides information on the tax data you have loaded into Oracle Receivables and Oracle Government Receivables or are reviewing for an upload. The report displays each senior location on a separate page. In the figure above, the state of California is the senior location.

The report is divided into the following sections:

- **Summary Section:** The summarized section of the Tax Interface report lists each action code that the Sales Tax Rate Interface program has used for the upload, the action code description, and the number of tax interface rows on which the program has executed each action.

- **Warning Section:** The warning section of the Tax Interface report lists every line of tax interface data that the Sales Tax Rate Interface program has identified with an action code other than 'INSERT' or 'NEW-LOCATION-INSERT'. Each row in this section lists the interface line ID, segment qualifier, location, postal code range, active date range, tax rate, and action for each row in the tax interface table that has a warning action code.
- **Locations and Rates Defined for Warnings Section:** This section of the report lists all of the locations and tax rates which are defined in your Oracle Receivables application for each location in the Warnings Section. This section includes the postal code and active date ranges of each of these locations' tax rates. Use this section in combination with the Warnings Section of the report to determine whether you need to take further action. If the tax rates are correct, then no further actions are necessary. If they are not, then you may need to update tax rates and locations manually in the Define Tax Locations and Rates form of your Oracle Receivables application. For example, if the Warnings Section of this report includes a tax interface line with an action of 'ALREADY-EXISTS' for a tax rate, then the Locations and Rates Defined for Warning section will show a tax rate for this location with the same postal code and active date range that already exists in your Oracle Receivables application. No action is necessary in this case.
- **Orphan Records, Parent_Location_ID Was Invalid Section:** The final section on the Tax Interface report lists lines of tax interface data which the Sales Tax Rate Interface program has identified as not having a valid parent location ID value. All locations that are junior to other locations identify their parent location through the parent location ID. The program will not load a tax record into your Oracle Receivables application if it cannot identify a parent location using this ID, or it is not a senior location. If this section of the report includes a tax interface line that is missing a Parent ID value, you must use SQL*Plus to update the line with the appropriate parent location ID, before it can be loaded into your Oracle Receivables application.

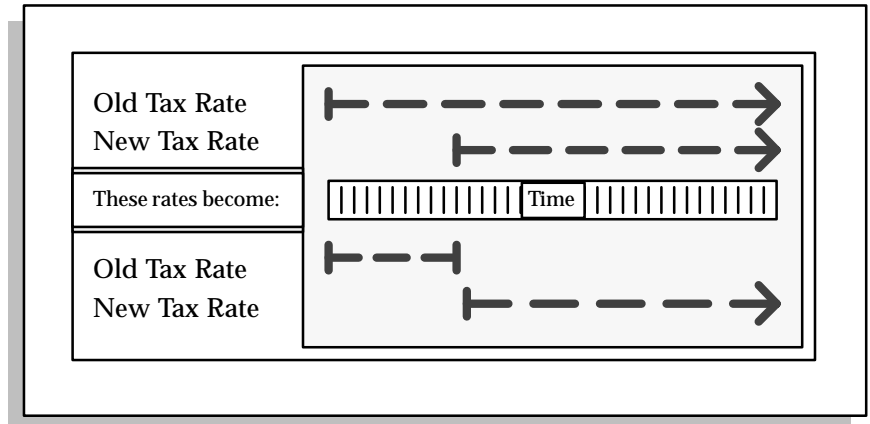
Sales Tax Rate Date Adjustments

Overview

The Sales Tax Rate Interface program adjusts the active date ranges of existing tax rates to ensure that they do not overlap with incoming tax rates for a location. For example, if an incoming sales tax rate for a location has start and end dates that overlap the active date range of an existing rate, then the Sales Tax Rate Interface program adjusts the active date range of the existing rate so that its active period does not overlap the date range of the incoming rate.

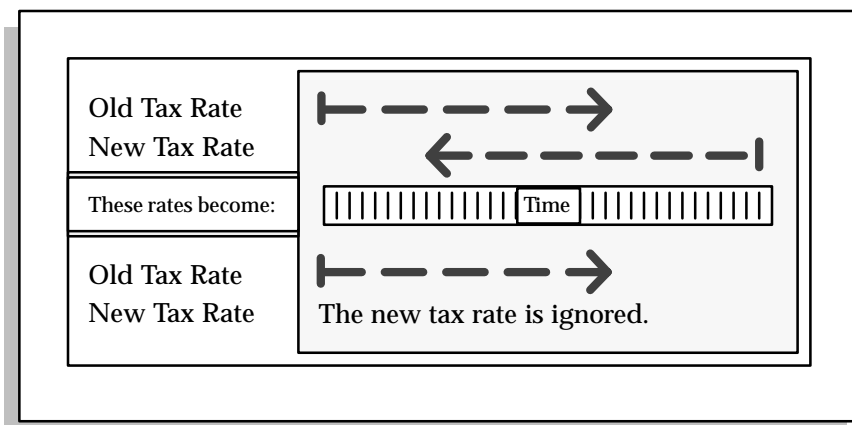
The following diagrams indicate how the Sales Tax Rate Interface program inserts new tax rates and updates the active date range of existing tax rates. Each diagram shows the action code associated with the type of rate date adjustment that the Sales Tax Rate Interface program performs.

Figure 10 – 10
UPDATE-INSERT



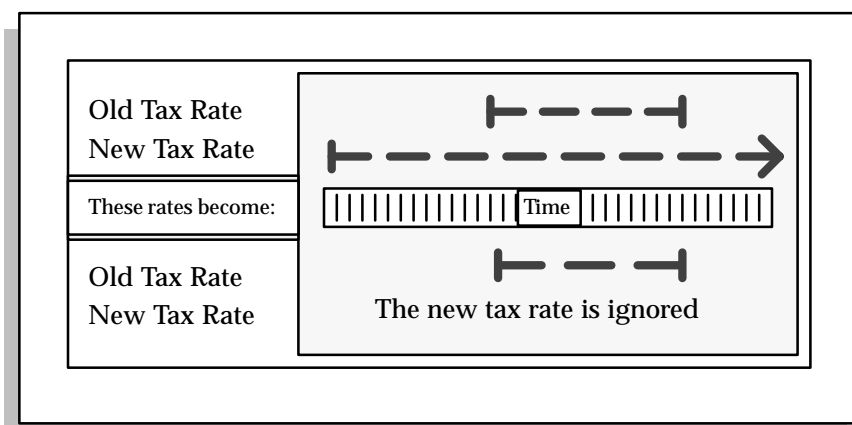
In Figure 10 – 10, when the Sales Tax Rate Interface program has assigned an action code of 'UPDATE-INSERT' to a new tax rate record, it has updated the existing tax rate by assigning it an end date immediately before the start date of the new tax rate. The program then inserts the new sales tax rate into your Oracle Receivables application.

Figure 10 – 11
IGNORE



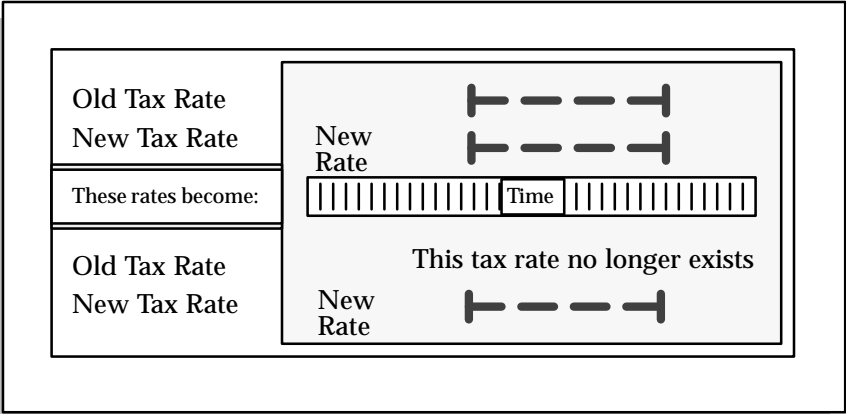
In Figure 10 – 11, when the Sales Tax Rate Interface program has assigned an action code of 'IGNORE' to a new tax rate record, it has ignored the new tax rate because it cannot determine either an end date for the existing rate, or a start date for the new rate. The existing tax rate remains unchanged.

Figure 10 – 12
IGNORE



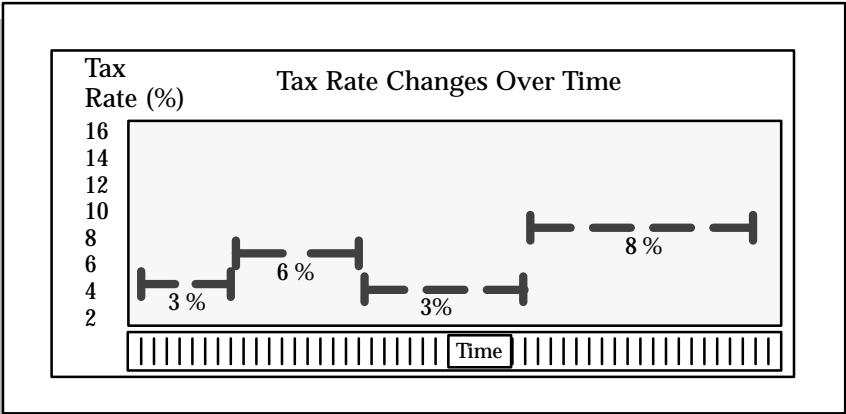
In Figure 10 – 12, when the Sales Tax Rate Interface program has assigned an action code of 'IGNORE' to a new tax rate record, it has ignored the new tax rate because it cannot determine an end date for this rate. The existing tax rate remains unchanged.

Figure 10 – 13
RATE-ADJUST



In Figure 10 – 13, when the Sales Tax Rate Interface program has assigned an action code of 'RATE-ADJUST' to a new tax record, it has replaced an existing tax rate with a new rate for the same location and active date range. You can either accept this new rate, or update it manually through the Define Tax Locations and Rates form in your Oracle Receivables application.

Figure 10 – 14
SALES TAX RATES
OVER TIME



As you continually update your tax rates using the Sales Tax Rate Interface program, you will store different tax rates for the same location that have mutually exclusive active date ranges. Figure 10 – 14 shows how your Oracle Receivables application will record different tax rates for a location as you update its rates over time.

Action Codes of Sales Tax Interface Program

The following lists include all of the possible actions that the Sales Tax Rate Interface program can perform on a row of tax data in the tax interface table.

Success Action Codes

The following codes are for tax rows that the Sales Tax Rate Interface program has successfully loaded into the location or tax rate tables. These codes do not require that you take any additional actions.

Action Code	Meaning
INSERT	Inserted a new tax rate for a location that already exists
NEW-LOCATION-INSERT	Inserted new location and new tax rate information

Table 10 – 25 Success Action Codes (Page 1 of 1)

Warning Action Codes

The following codes are warning action codes. You should verify that the Sales Tax Rate Interface program has performed the updates or adjustments that you require, for each record with a warning action code. For example, if a row of tax data from the tax interface table receives an action of 'RATE-ADJUST', then you should verify that the new tax rate for the location associated with this tax rate is correct.

Action Code	Meaning
ALREADY-EXISTS	This tax rate already exists in Oracle Receivables and Oracle Government Receivables
RATE-ADJUST	Replaced an existing tax rate with a new tax rate
UPDATE-INSERT	Inserted new tax rate information, and adjusted the active dates of existing tax rate information

Table 10 – 26 Warning Action Codes (Page 1 of 1)

Error Action Codes

The following codes indicate that an Oracle error has occurred during the processing of tax data by the Sales Tax Rate Interface program. The program will still produce the Tax Interface report even if an Oracle error occurs during the upload process.



Oracle Message Reference Manual

Action Code	Meaning
ORACLE_ERROR	An Oracle error has occurred. The Oracle error number is listed for this error.

Table 10 – 27 Error Action Codes (Page 1 of 1)

The following action codes indicate that either the tax data in the tax interface table is invalid, or the Sales Tax Rate Interface program did not interpret this data correctly. Although the Sales Tax Rate Interface program may assign these codes to records in the tax interface table, these codes should not appear on the Tax Interface report in normal circumstances.

Note: These codes would, however, appear in the report if you elected to run it in review mode, immediately after using SQL Loader to load your data, but before uploading into your Oracle Receivables application.

Action Code	Meaning
IGNORE	The Sales Tax Rate Interface program did not upload this record.
<NULL>	This record is ready to be loaded by the Sales Tax Rate Interface program.

Table 10 – 28 Error Action Codes (Page 1 of 1)

Truncated City Names

Your Oracle Receivables application tax forms and QuickPicks support up to 60 characters for each location. Your tax feeder system may support a smaller number of characters. For example, if you have a State, County and City name of Arizona, Havasupai, Havasupai Indian Reservation, and your Sales Tax feeder system only supports 25 characters for each name, the city named Havasupai Indian Reservation is truncated to "Havasupai Indian Reservat". You must update any truncated city names in the Define Tax Locations and Rates form to reflect the correct names.

Sales Tax Rate Interface Program Updates of AR_TAX_INTERFACE Table

Below is a description of the columns in the tax interface table AR_TAX_INTERFACE which the Sales Tax Rate Interface program updates during execution. All other columns in this table remain unchanged after the initial load from your feeder system. You can review this data by running the Sales Tax Rate Interface report in Review mode.

Column Name	Value
STATUS	The Sales Tax Rate Interface program populates this column with the action code that it identifies for this record.
LOCATION_SEGMENT_ID	The Sales Tax Rate Interface program populates this column with the ID of the location associated with this row. This ID column links a location in your Oracle Receivables application to the rows of tax

interface data that relate to this location, and is a foreign key to AR_LOCATION_VALUES.LOCATION_SEGMENT_ID.

Note: If you need to reload the tax data for a month that you have previously loaded into the interface tables, then you should first null the above two columns to reset the data. You can then run the sql*loader file again and reload the data for that month. If you do not null these two columns then Oracle Receivables will not reload the same month's data.

Termination of the Tax Interface Report

If more than one hundred Oracle errors occur during the upload of a single senior location, the program will terminate with an error message indicating that too many errors have occurred. Because the program saves after uploading the tax records of each senior location value, the Tax Interface report lists all of the tax records that the program has successfully uploaded, as well as the Oracle errors which resulted in the termination of the program. All of the remaining tax records of senior locations that the Sales Tax Rate Interface program has not processed will have an status of <NULL>.

If you resubmit the Sales Tax Rate Interface program using the same parameter values, the program only processes those tax records that satisfy these criteria and have an action code status of <NULL>. If the Tax Interface report indicates that the Oracle errors only occurred on the tax data of a specific senior location, then you should specify this location as your senior location parameter when you resubmit the Sales Tax Rate Interface program.

Oracle Receivables and Oracle Government Receivables Tables

The Sales Tax Rate Interface program transfers sales tax data into the following Oracle Receivables and Oracle Government Receivables tables:

AR_LOCATION_VALUES

AR_LOCATION_RATES

Table and Column Descriptions

AR_TAX_INTERFACE

Below is a description of the columns in the AR_TAX_INTERFACE table into which you load sales tax data using SQL*Loader.

The Sales Tax Rate Interface program transfers the sales tax rate data from AR_TAX_INTERFACE to the AR_LOCATION_VALUES and AR_LOCATION_RATES tables. These two tables store your locations and tax rates respectively. The AR_TAX_INTERFACE table's column descriptions note the columns in the AR_LOCATION_VALUES and AR_LOCATION_RATES tables into which records are transferred. If the column description does not have a destination section, then this information will not be transferred.

Column Name	Value
REQUEST_ID	The concurrent request ID of each submission of the Sales Tax Rate Interface program. Your Oracle Receivables application allows you to submit multiple uploads of tax data concurrently. This column is not null.
INTERFACE_LINE_ID	The sequence number that uniquely identifies each tax data line. This column is not null.
CREATED_BY	The ID of the Oracle Application Object Library user who has loaded this tax data record into this table. This column is not null.
CREATION_DATE	The date on which the user loaded this tax data record into this table. This column is not null.
SEGMENT_QUALIFIER	Identifies which segment this tax data record is for. 'COUNTRY', 'STATE', 'COUNTY', 'CITY', and 'PROVINCE' are examples of possible segment qualifier values. This column is not null. Destination: AR_LOCATION_VALUES.LOCATION_SEGMENT_QUALIFIER
RATE_TYPE	Identifies the tax rate type of this tax data record. Will always be 'SALES'. This column is not null.

LOCATION_ID	The ID (or Geo-code) for this location. This ID number must be unique for every location. This number is composed of a unique identifier for the location prefixed by the parent location ID. This column is not null.
LOCATION_VALUE	<p>The actual location. 'CA' is an example of a location for the segment of 'STATE'.</p> <p>Destination: AR_LOCATION_VALUES.LOCATION_SEGMENT_VALUE</p>
LOCATION_DESCRIPTION	<p>The description of the location. For a location of 'CA', the description is 'California'.</p> <p>Destination: AR_LOCATION_VALUES.LOCATION_SEGMENT_DESCRIPTION</p>
PARENT_LOCATION_ID	The ID of the location which is directly senior to this record. This value is a foreign key to the LOCATION_ID column in the tax interface table.
FROM_POSTAL_CODE	<p>The starting postal code for the postal code range of this tax rate. If this column is null, your Oracle Receivables application uses the value that you have defined for the Postal Code Range: From field in the Define System Options form for this column. This value is stored in the column AR_SYSTEM_PARAMETERS.FROM_POSTAL_CODE</p> <p>Destination: AR_LOCATION_RATES.FROM_POSTAL_CODE</p>
TO_POSTAL_CODE	<p>The ending postal code for the postal code range of this tax rate. If this column is null, your Oracle Receivables application uses the value that you have defined for the Postal Code Range: To field in the Define System Options form for this column. This value is stored in the column AR_SYSTEM_PARAMETERS.TO_POSTAL_CODE.</p> <p>Destination: AR_LOCATION_RATES.TO_POSTAL_CODE</p>
START_DATE	<p>The start date of a tax rate for this location. If this column is null, then your Oracle Receivables application uses the value '01-JAN-1900 00:00:00' for this column.</p> <p>Destination: AR_LOCATION_RATES.START_DATE</p>
END_DATE	<p>The end date of a tax rate for this location. If this column is null, then your Oracle Receivables application uses the value '31-DEC-2199 23:59:59' for this column.</p> <p>Destination: AR_LOCATION_RATES.END_DATE</p>
TAX_RATE	The tax rate for this location.

Destination: AR_LOCATION_RATES.TAX_RATES

CHANGE_FLAG

A flag which indicates if this tax record is different from the previous listing of this data. Set this column to 'Y' if this record has been changed, 'N' if it has not, and 'U' if it is unknown whether this tax record has changed. This column is not null.

Implementing the Oracle Receivables Applications Tax Vendor Extension

Oracle Receivables and Oracle Government Receivables provide a Tax Vendor Extension to integrate external tax calculation programs with Oracle Applications. This Extension enables you to provide for complex tax calculation needs, while retaining the full power of your Oracle Receivables application to create and store all other tax data.

The Tax Extension is called whenever a tax rate is calculated by the your Oracle Receivables application tax engine. When implemented, the Tax Extension will return a tax rate or amount from the vendor program. Your Oracle Receivables application will use this information to create the appropriate tax line(s) and related accounting information.

Basic Needs

Your Oracle Receivables application provides you with the following features to calculate tax amounts from external programs:

- Integrate with vendors of tax rate calculation programs
- Calculate flat tax amounts
- Calculate multiple tax lines
- Calculate compound tax lines based on precedence numbers
- Calculate tax amounts using a set of parameters that you define

These include:

Ship From/To Address

Point of Title Passage

Point of Order Acceptance

Point of Order Origin

Inventory Item

Quantity

Amount

Transaction Date

- Calculate tax amounts rounded to a specific precision and minimum accountable unit
- Integrate your Tax Vendor with Oracle Order Entry

Major Features

Sales Tax Rate Interface

Both Oracle Order Entry and your Oracle Receivables application provide an interface to load tax rates, zip code ranges and location names from data files supplied by external vendors. This information is used by the Enter Customer Information and Quick Customer Entry forms to validate addresses and create compiled sales tax data, whenever an address is updated or created.

Two sample SQL*Loader control files aravp.ctl and arvertex.ctl, are provided. These sample SQL*Loader control files will also load jurisdiction code data into your Oracle Receivables application AR_LOCATION_RATES table. Once loaded, the jurisdiction code data can be referenced by customizations to your tax database views.

Tax Database Views

Your Oracle Receivables application provides database views to control which database columns are passed into the tax engine to calculate tax for your transaction lines.

You can modify these views to optimize the proprietary features of your external tax calculation program. These features may allow for more accurate tax calculations.

Generic Tax Extension

The Tax Extension is called whenever a tax rate is calculated by your Oracle Receivables application tax engine. Tax rates are calculated in the following forms and concurrent programs:

- AutoInvoice
- Recurring Invoices
- Invoice Entry
- Enter Orders (Oracle Order Entry)



Warning: The Tax Extension is not called from the Credit Memo or Adjustment forms.

The Tax Extension can be implemented to generate single or multiple tax lines for every invoice line. Your Oracle Receivables application will store each tax line in the RA_CUSTOMER_TRX_LINES table.

AutoAccounting will generate one account distribution for each tax line returned from the Tax Vendor and your Oracle Receivables application will post this information to the General Ledger.

Calculating Multiple Tax Lines and Compound Taxes

The tax database views can be modified to return multiple tax lines per invoice line. This is useful for example, if you need to create tax lines for each component of your Sales Tax Location Flexfield, rather than having a total rate for all the components.

Additionally, if you elect to return multiple tax lines from the views, you can assign precedence numbers to indicate which tax lines you want to compound. Your Oracle Receivables application will use these precedence numbers to calculate a compound tax, based on the rates and/or amounts returned by your Tax Vendor for each tax line.

Integration With Oracle Order Entry

The Oracle Receivables and Oracle Government Receivables Tax Extension is fully integrated with Oracle Order Entry. At the time of order entry the tax amount for an order or line is calculated by calling your Oracle Receivables application tax engine. Consequently, if you have installed a Tax Vendor, it will be called to calculate tax on the order in the same way as on the invoice.

Note: Tax on an order is for information only and will be recalculated at the time of invoice creation. This is necessary because tax rates change over time and there could be a large gap between when an order is taken and when the customer is invoiced for the items on this order.

Definitions

Tax Engine

The tax engine is a collection of programs, user defined system parameters, and hierarchical flows used by your Oracle Receivables application to calculate tax.

Minimum Accountable Unit

Minimum Accountable Unit is the smallest meaningful denomination of a currency and might not correspond to the standard precision.

While a currency may require a precision of three places to the right of the decimal point, .001(one thousandth), the lowest denomination of the currency may represent 0.025 (twenty-five thousandths). Under this example the Minimum Accountable Unit would be .025. Calculations in this currency would be rounded to .025 (the Minimum Accountable Unit) not .001 (the precision).

Flat Tax Amounts

A Flat Tax is a specific amount of tax, regardless of the amount of the item. There is no rate associated with flat taxes. Flat taxes are charged on items such as cigarettes, gasoline and insurance.

Jurisdiction Code

A jurisdiction code is an abbreviated address that is specific to a Tax Vendor and more accurate than a simple five digit zip code.

Precedence Numbers

Precedence Numbers are used to determine how your Oracle Receivables application will compound taxes. The tax line with the highest precedence number will calculate tax on all tax lines with a lower precedence number.

Compound Tax

Compound Tax is a tax based on other taxes. It is calculated when other tax amounts are included in the taxable basis of the item to be taxed.

Preparing Oracle Receivables and Oracle Government Receivables

The Tax Extension is a public C function, which is called by the Oracle tax engine every time a tax rate is calculated within your Oracle Receivables application or Oracle Order Entry.

This function is named `artaxvdr()` and has four associated files:

Directory	File Name	Description
AR_TOP/usrxit/	artaxvdr.lpc	Source Code
AR_TOP/usrxit/	artaxvdr.h	Declares Data Structures
AR_TOP/usrxit/	artxview.sql	Tax Database Views
AR_TOP/usrxit/	readme.txt	Developer Notes

Table 10 – 29 (Page 1 of 1)



Suggestion: Review these files before you implement the Tax Extension. They provide a lot of useful information.

If a Tax Vendor is not installed, the Tax Extension will pass back the 'No Vendor' return code and your Oracle Receivables application will use the tax rates calculated internally. To install your Tax Vendor you must modify the Tax Extension.

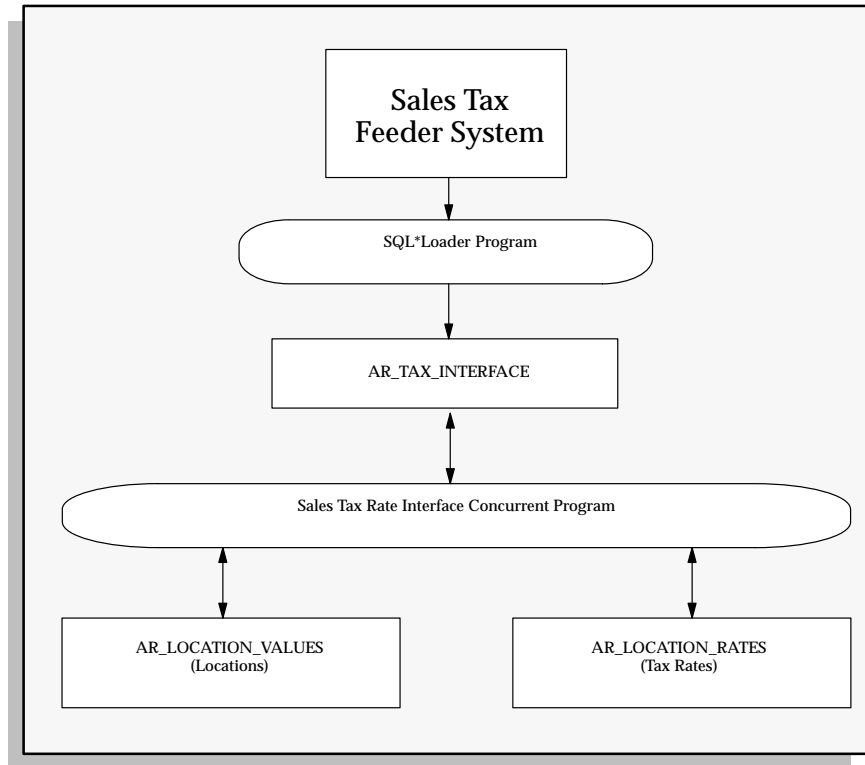
Additionally, you can perform the following optional steps to prepare your Oracle Receivables application for the calculation of tax on invoice lines, by an external Tax Vendor.

Load External Tax Information

Your Oracle Receivables application provides two sample SQL*Loader control files, `AR_TOP/bin/aravp.ctl` and `AR_TOP/bin/arvertex.ctl`, to load new locations and tax rates from data files supplied by your Tax Vendor. These programs allow you to load multiple tax rates for the same location, which may cover different date ranges and postal codes. The following diagram shows how your Tax Vendor's data is imported into your Oracle Receivables application tax tables.

Note: Your Oracle Receivables application provides six possible Sales Tax Location Flexfield structures. The sample SQL*Loader files, `aravp.ctl` and `arvertex.ctl`, only support the structure, `State.County.City`. If you select another structure, you will have to modify these SQL*Loader files.

Figure 10 – 15Sales Tax Rate Interface



Using Sales Tax Rate Interface (See page 10 – 198)

Modify the Database Views

Your Oracle Receivables application provides database views to pass tax information to the tax engine. You can use these views to control which database columns are passed into the tax engine for every transaction line you tax.

The following views have been defined.

- **SO_TAX_LINES_SUMMARY_V**
Used by the Order Entry form.
- **SO_TAX_LINES_CREDIT_CHECK_V**
Used by the Order Entry form.
- **TAX_LINES_INVOICE_IMPORT_V**
Used by the AutoInvoice program.
- **TAX_LINES_RECREATE_V**

Used by the Enter Invoices form.

- TAX_LINES_INQUIRY_V

Used by the Enter Invoices form.

- TAX_VENDOR_JURISDICTIONS_V

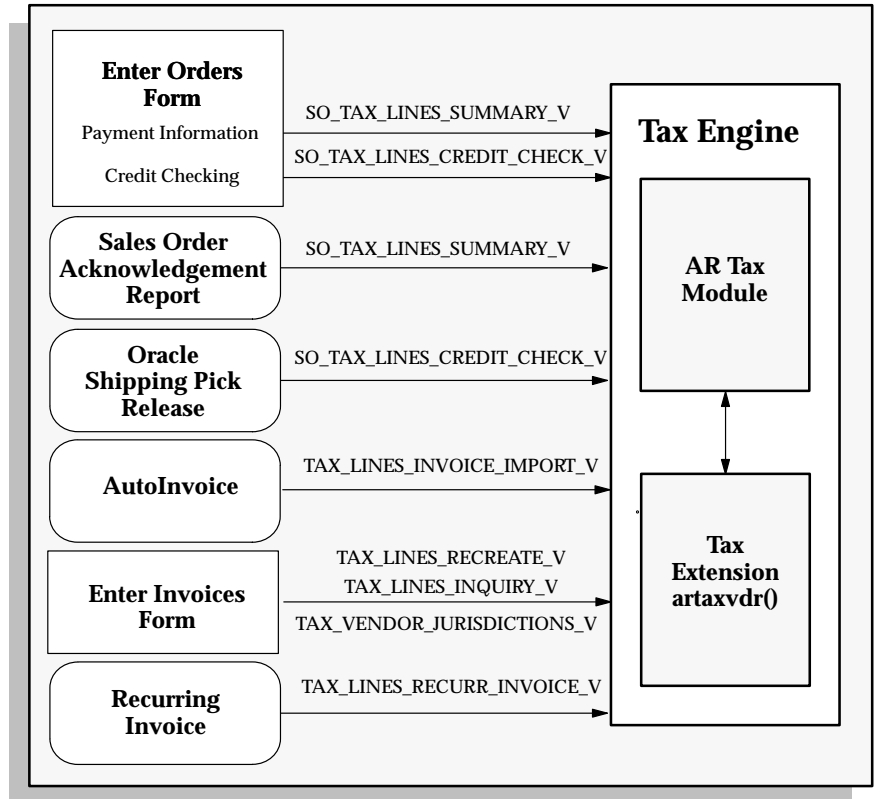
Used by the Enter Invoices form.

- TAX_LINES_RECURRENCE_INVOICE_V

Used by the Recurring Invoice program.

For more information please refer to the **Oracle Receivables Application Technical Reference Manual** and the **Oracle Order Entry Application Technical Reference Manual**.

Figure 10 – 16



The Tax Engine uses the information passed by the database views to calculate tax, regardless of whether an external Tax Vendor is installed. Both the Tax Extension and the AR Tax Module are components of the Tax Engine and are called every time the Tax Engine is requested to calculate tax. If an external Tax Vendor is installed, the Tax Engine will

use the tax rate or amount returned by the Tax Extension to override the rate or amount calculated by the AR Tax Module.

If the following columns, available in each view, are not populated, the Tax Extension will be passed NULL for each of their values.

View Column Name	Description
POO_ADDRESS_CODE	Jurisdiction code for Point of Order Origin
POA_ADDRESS_CODE	Jurisdiction code for Point of Order Acceptance
SHIP_FROM_ADDRESS_CODE	Jurisdiction code for Ship From Address
SHIP_TO_ADDRESS_CODE	Jurisdiction code for Ship-To Address
FOB_CODE	Free On Board or Point of Title Passage
PART_NUMBER	Inventory Part Number
LOCATION_QUALIFIER	Identifies the Taxing Authority, e.g. 'ALL', 'STATE', 'COUNTY' or 'CITY'

Table 10 – 30 (Page 1 of 1)

Implementing Tax Jurisdictions

Within the United States a tax rate is calculated from Ship-To, Ship From, Point of Order Origin and Point of Order Acceptance. To implement the Tax Extension using each of these addresses, you will have to store the latter three values in descriptive flexfields at the appropriate level; Invoice Line or Header or Order Line or Header.

If you use AutoInvoice to import orders from Oracle Order Entry, AutoInvoice will populate the item line Transaction Flexfield with packing slip information. This may be used to source the Ship From site use and address for each order.



Integrating Oracle Order Entry with Oracle Receivables
(*Oracle Financials and Oracle Government Financials Open Interfaces Manual*)

Following is an example of how to modify the views to select vendor jurisdiction codes into the SHIP_TO_ADDRESS_CODE column:

```
CREATE or REPLACE view TAX_VENDOR_JURISDICTIONS_V as
SELECT      rct.CUSTOMER_TRX_ID,
            rct.TRX_NUMBER,
            decode(a_ship_to.location_id, NULL,
                  decode(a_ship_to.country, 'US', '**ERROR**',
                        NULL,NULL, '**INTERNATIONL**'),
                  decode(&TAX_VENDOR, 'AVP',
                        upper(substr(a_ship_to.state,1,2)) ||
                        substr(a_ship_to.postal_code,1,5) ||
                        nvl(a_ship_to.attrbiutel,
                            nvl(alr_ship_to.attribute1, '00')),
                        'VERTEX',
                        nvl(a_ship_to.attrbiutel,
                            nvl(alr_ship_to.attribute1,
                                '**BAD ADDRESS**'))))
            ship_to_address_code,
            ...
FROM        RA_CUSTOMER_TRX          rct,
            RA_ADDRESSES             a_ship_to,
            AR_LOCATION_RATES        alr_ship_to,
            AR_LOCATION_COMBINATIONS alc_ship_to,
            RA_SITE_USES             s_ship_to,
            ...
WHERE       ...
```

Note: A Tax Vendor's Jurisdiction Code can be assigned to the address manually using `ra_addresses.attribute1`; or uploaded automatically from the Tax Vendor's data file using `ar_location_rates.attribute1`.

```
AND      s_ship_to.SITE_USE_ID(+) = rct.SHIP_TO_SITE_USE_ID
AND      a_ship_to.ADDRESS_ID(+) = s_ship_to.ADDRESS_ID
AND      a_ship_to.LOCATION_ID = alc_ship_to.location_id(+)
AND      alc_ship_to.location_id_segment_3 =
            alr_ship_to.location_segment_id(+)
AND      decode(alr_ship_to.location_segment_id,
            null, '00000', a_ship_to.postal_code)
            between nvl(alr_ship_to.from_postal_code, '00000')
            and nvl(alr_ship_to.to_postal_code, '00000')
AND      rct.trx_date
            between nvl(alr_ship_to.start_date, rct.trx_date)
            and nvl(alr_ship_to.end_date, rct.trx_date)
            ...
```


The jurisdiction codes are loaded by the Sales Tax Rate Interface into attribute 1 of the table *ar_location_rates*. To load vendor jurisdiction codes into the other view columns, you will have to modify the views to join *ar_location_rates* with your appropriate customized table.

If you require postal code data to nine characters (zip+4) to segregate customer addresses by jurisdiction code, you will have to manually update the address data provided by your Tax Vendor. You can use the "Define Location and Rates" form to update the postal code data to comply with your jurisdiction code requirements. Below is an example of multiple jurisdiction codes within a standard five digit zip code designation:

location_segment_id	from_postal_code	to_postal_code	jurisdiction code
43 (San Francisco)	94110	94116	code 1
43 (San Francisco)	94117	94117	code 2
43 (San Francisco)	94118	94118-3999	code 3
43 (San Francisco)	94118-4000	94118-9999	code 4

Table 10 – 31 (Page 1 of 1)

Generating Multiple Tax Lines

To generate multiple tax lines per invoice line, the views would be changed to return multiple rows per invoice line, with each line identified by a unique location_qualifier. For example:

TRX_ID	TRX_LINE_ID	LINE_NO	EXTENDED _AMOUNT	LOCATION_QUALIFIER
123	78955	1	5000	State
123	78955	1	5000	County
123	78955	1	5000	City
123	78955	1	5000	Federal
123	78955	1	5000	Provincial

Table 10 – 32 (Page 1 of 1) Example of Multiple Tax Lines per transaction line.

Your Oracle Receivables application would then store each tax rate as a separate tax line on the invoice.



Warning: Be careful to use outer joins for your customizations of the tax database views. Failure to do so may result in order or invoice lines not being taxed.

Installing the Tax Extension

To install your Tax Vendor you must modify the C source file, *artaxvdr.lpc*, to call your Tax Vendor and pass the tax engine the appropriate return code and tax rate or amount.

The following section provides an example of how to modify the source code to implement the hook.

Tax Extension Parameters

The Tax Extension function takes four parameters:

Parameter	Description
ub4 event	Event code
t_artaxline *artaxline	Tax calculation parameters
dvoid *arg_struct	Internal pointer used for error reporting
dvoid *vdr_cache	Not used on most ports

Table 10 – 33 (Page 1 of 1)

- **PARAMETER:event**, indicates the type of operation the Tax Vendor is required to perform. The event parameter has one of the following values:

ARTAXVDR_STARTUP

Used to initialize your Tax Vendor, allocate memory for your own applications and read any required profile values. The Tax Extension will be passed the Start Up event for every connection to Oracle Applications. If you have installed a Tax Vendor, the function will return the code of 'Success', otherwise the function will return the code of 'No Vendor'.



Warning: You or your Tax Vendor should not attempt to connect to the database within this event. Oracle Applications will have already connected you to the appropriate architecture based on your responsibility and application.

ARTAXVDR_RESET

Used to clear out information held by your Tax Vendor across calls. A Reset event will be passed to the Tax Extension for every request from a form to the tax engine.

ARTAXVDR_CALCULATE

Used to compute a tax rate for each row returned from the database views. The Tax Extension will be passed an event of Calculate for every request from a form or concurrent program to the tax engine.

ARTAXVDR_DELETE

Used to reverse existing audit records in the Tax Vendor files. A Delete event will be passed to the Tax Extension for every request to delete tax lines.

ARTAXVDR_RELEASE

Used to close your Tax Vendor and release any allocated memory. The Tax Extension will be passed the Release event for each connected session to Oracle Applications. If you have installed a Tax Vendor, the function will return the code of 'Success', otherwise the function will return the code of 'No Vendor'.

- **PARAMETER:artaxline**, a pointer to the data structure in the file *artaxvdr.h*, which lists all the data members of a transaction line that is to be taxed. Each view column maps to a different member of the data structure.

The Tax Extension data structure has three types of members,

1) Input members are defined by the database views and represent the data to be passed to your Tax Vendor from your Oracle Receivables application.

Member	Data Type	Parameter Type	Null Allowed	Related View Column
bill_to_cust_id	ub4	input	no	CUSTOMER_ID
ship_to_cust_id	ub4	input	yes	CUSTOMER_ID
trx_id	ub4	input	yes	TRX_ID
trx_line_id	ub4	input	yes	TRX_ID_LINE
trx_date	text	input	yes	TRX_DATE

Table 10 – 34 (Page 1 of 3) Input members of the data structure, artaxvdr.h

Member	Data Type	Parameter Type	Null Allowed	Related View Column
item_flex_id	ub4	input	yes	INVENTORY_ITEM_ID
quantity	double	input	yes	QUANTITY
amount	double	input	yes	EXTENDED_AMOUNT
orgid	ub4	input	no	INVENTORY_ORG_ID
sobid	ub4	input	no	SET_OF_BOOKS_ID
audit_flag	boolean	input	no	auditable transaction
line_status	ub4	input	no	default return code
exchange_rate	double	input	yes	EXCHANGE_RATE
currency_code	text	input	yes	CURRENCY_CODE
poo_code	text	input	yes	POO_CODE
poa_code	text	input	yes	POA_CODE
ship_from_code	text	input	yes	SHIP_FROM_CODE
ship_to_code	text	input	yes	SHIP_TO_CODE
fob	text	input	yes	FREE_ON_BOARD
part_no	text	input	yes	INVENTORY_PART_NUM- BER
qualifier	text	input	no	LOCATION_QUALIFIER
calculate_tax	text	input	yes	TAXABLE_FLAG
vdrctrl_exempt	text	input	yes	VENDOR_CONTROL _EXEMPTIONS
userf1	text	input	yes	ATTRIBUTE1
userf2	text	input	yes	ATTRIBUTE2
userf3	text	input	yes	ATTRIBUTE3
userf4	text	input	yes	ATTRIBUTE4
userf5	text	input	yes	ATTRIBUTE5
usern1	double	input	yes	NUMERIC_ATTRIBUTE1
usern2	double	input	yes	NUMERIC_ATTRIBUTE2
usern3	double	input	yes	NUMERIC_ATTRIBUTE3

Table 10 – 34 (Page 2 of 3) Input members of the data structure, artaxvdr.h

Member	Data Type	Parameter Type	Null Allowed	Related View Column
usern4	double	input	yes	NUMERIC_ATTRIBUTE4
usern5	double	input	yes	NUMERIC_ATTRIBUTE5

Table 10 – 34 (Page 3 of 3) Input members of the data structure, artaxvdr.h

2) Input/Output members represent optional tax information. If you elect to maintain your tax codes and exemption information within your Oracle Receivables application, you will have to modify the database views to pass this information to the Tax Extension. The Tax Extension will then pass this data to your Tax Vendor. If you elect to use your Tax Vendor's exemption and tax code data, you may pass these members a null value from the database views. If these members are null, your Tax Vendor will use its own exemption data to calculate your tax and will return these values accordingly.

Member	Data Type	Parameter Type	Null Allowed	Related View Column
tax_code	text	input/output	yes	TAX_CODE
vat_tax_id	sb4	input/output	yes	VAT_TAX_ID
tax_rate	double	input/output	yes	TAX_RATE
minimum_accountable_unit	double	input/output	yes	MINIMUM_ACCOUNTABLE_UNIT
precision	double	input/output	yes	PRECISION
tax_exempt_flag	text	input/output	yes	"E" "S" or "R"
item_exception_rate_id	ub4	input/output	yes	ITEM_EXCEPTION_RATE_ID
tax_exemption_id	ub4	input/output	yes	TAX_EXEMPTION_ID
xmpt_cert_no	text	input/output	yes	TAX_EXEMPT_NUMBER
xmpt_reason	text	input/output	yes	TAX_EXEMPT_REASON
xmpt_percent	double	input/output	yes	TAX_EXEMPT_PERCENT

Table 10 – 35 (Page 1 of 1) Input/Output members of the data structure, artaxvdr.h

3) Output members, which hold the data passed back to the Tax Extension by your Tax Vendor.

Member	Data Type	Parameter Type	Null Allowed	Related View Column
o_tax_amount	double	output	no*	
o_taxvdr_rc	text	output	no	
precedence	double	output	yes	

Table 10 – 36 (Page 1 of 1) Output members of the data structure, artaxvdr.h

* o_tax_amount is null allowed if tax_rate is set.

- **arg_struct**, a pass through pointer. It is used internally by Oracle Applications when reporting errors.
- **vdr_cache**, a tool to implement static data structures on platforms which do not support static data structures. This parameter is set during start up and is returned upon every subsequent call to the function.

Note: This parameter can be ignored if your hardware platform supports static variables, which nearly every platform supports.

Tax Extension Return Codes

The default return code for the Tax Extension function will be 'No Vendor' and the tax will be calculated internally by the tax engine. When a Tax Vendor is installed, the Tax Extension can receive the following return codes:

- **ARTAXVDR_SUCCESS**

The external tax calculation program has returned an appropriate tax rate or amount. The calling program or form will continue processing without alerting the user.

- **ARTAXVDR_NO_RATE**

The external tax calculation program could not find an appropriate tax rate for the given transaction line. This is an error return code and if called from a form, your Oracle Receivables application will report an error message, which must be accepted to continue processing. If the Tax Extension is called from a concurrent program, the calling program will not generate tax lines for the erroneous transaction, will mark the transaction as not having tax lines and then continue processing other transactions.

Note: If the profile AR: Allow Manual Lines is enabled, the user could manually enter a tax line or add additional tax lines even if the Tax Extension could not calculate a tax.

- **ARTAXVDR_NO_POSTAL_CODE**

The external tax calculation program could not calculate a tax amount because it does not have a correct postal code for the given transaction line. This is an error return code and if called from a form, your Oracle Receivables application will report an error message, which must be accepted to continue processing. If the Tax Extension is called from a concurrent program, the calling program will not generate tax lines for the erroneous transaction, will mark the transaction as not having tax lines and then continue processing other transactions.

- **ARTAXVDR_OERR**

An Oracle error has occurred during the execution of the event and the calling program will abort. You should only use this return code if further processing of transactions is inappropriate. Before returning this error code you should generate an error message by calling the function `artaxerm()`, discussed below.

- **ARTAXVDR_SYSERR**

A system or memory management error occurred during the execution of the event, which will abort the calling program. You should only use this return code if further processing of transactions is inappropriate. Before returning this error code you should generate an error message by calling the function `artaxerm()`, discussed below.

- **ARTAXVDR_NO_VENDOR**

The default return code.

Modify the Tax Extension

The following C code provides an example of how to modify the Tax Extension function, `artaxvdr()`, within the events `artaxvdr_startup`, `artaxvdr_calculate` and `artaxvdr_release`.

```
#ifndef ARTAXVDR_ORACLE
#include <artaxvdr.h>
#endif
#ifndef SQLCA
EXEC SQL INCLUDE SQLCA.H;
#endif
```

```

#include <vendor.h>
EXEC SQL BEGIN DECLARE SECTION;
EXEC SQL END DECLARE SECTION;
FDRCSID("$Header: artaxvdr.lpc Custom");
ub4 artaxvdr( event, arg_struct, vdr_cache, artaxline )
ub4 event;
dvoid *arg_struct;X
dvoid **vdr_cache;
t_artaxline *artaxline;
{
    ub4    return_code;
    text   vendor_error_message;
    type_vendor_struct vendor_struct;
    if ( event == ARTAXVDR_STARTUP )
    {
        /* ----- */
        /* Initialise Tax Vendor */
        /* ----- */
        if !(tax_vendor_start( &vendor_error_message))
        {
            artaxerrm(ARTAXVDR_EXPANDED_TEXT,
                      arg_struct,
                      vendor_error_message,
                      (sb4)0,
                      (text *)" " );
            return(ARTAXVDR_SYSERR);
        }
        return_code = ARTAXVDR_SUCCESS;
    }
else if (event == ARTAXVDR_RELEASE )
    {
        /* ----- */
        /* Closedown Tax Vendor */
        /* ----- */
        if !(tax_vendor_stop( &vendor_error_message))
        {
            artaxerrm( ARTAXVDR_EXPANDED_TEXT,
                      arg_struct,
                      vendor_error_message,
                      (sb4)0,
                      (text *)" " );
            return( ARTAXVDR_NO_RATE );
        }
    }
}

```



```

        return_code = ARTAXVDR_SUCCESS;
    }
else if ( event == ARTAXVDR_CALCULATE )
{
    /* ----- */
    /* Calculate a Tax Rate */
    /* ----- */

    /* If Called from within the Enter Invoice Form the*/
    /* line item record will not have been posted to   */
    /* the database yet. Therefore, jurisdiction        */
    /* columns for this transaction will be null.       */
    /* These jursdiction columns should be manually    */
    /* fetched from the database given the invoice     */
    /* header ID. This avoids customizing the Enter   */
    /* Invoices form. */

    if ((artaxline->i_ship_to_code == -1 ) &&
        (artaxline->i_bill_to_code == -1 ) &&
        (artaxline->i_ship_from_code == -1 ) &&
        (artaxline->i_ship_to_code == -1 ) &&
        (artaxline->i_part_no  == -1 ))
    {
        SELECT POO_CODE,POA_CODE, SHIP_FROM_CODE,
        SHIP_TO_CODE,
        INTO artaxline->poo_code, artaxline->poa_code, ....
        FROM TAX_VENDOR_JURISDICTIONS_VIEWS
        WHERE TRX_HEADER_ID = <:artaxline.trx_header_id>;

        SELECT attributeN
        INTO artaxline->part_no;
        FROM MTL_SYSTEM_ITEMS
        WHERE INVENTORY_ITEM_ID = <:artaxline.trx_header_id>;
    }

    /* Copy Parameters from artaxline into vendors data */
    /* structure. */
    strcpy(vendor_struct->poo, artaxline->poo_code );
    strcpy(vendor_struct->poa, artaxline->poo_code );
    strcpy(vendor_struct->shf, artaxline->poo_code );
    strcpy(vendor_struct->sht, artaxline->poo_code );
    strcpy(vendor_struct->product_code,artaxline->part_no);
    if (artaxline->i_xmpt_reason == -1 )
    {

```

```

        vendor_struct->exempt = 0; /* False */
    }
    else
    {
        vendor_struct->exempt = 1; /* True */
    }
    ...
    /* Call the Tax Vendor to Calculate a rate.Note that*/
    /* artaxline->qualifier will be used to name each    */
    /* of the required tax lines, i.e. STATE, COUNTY,    */
    /* CITY or ALL, i.e. STATE+COUNTY+CITY.            */
    /*                                                    */

    if !(tax_vendor_calc(&vendor_struct,
                        &vendor_error_message))
    {
        artaxerrm(ARTAXVDR_EXPANDED_TEXT,
                arg_struct,
                vendor_error_message,
                (sb4)0,
                (text *)"");
        return( ARTAXVDR_NO_RATE );
    }
    /* Mark records generated by Tax Vendor */
    strcpy(artaxline->o_taxvdr_rc, "ok");
    artaxline->io_taxvdr_rc = 0;

    /* Copy vendor_struct->output_rate into artaxline */
    artaxline->tax_rate = vendor_struct->tax_rate;
    artaxline->i_tax_rate = 0;

    return_code = ARTAXVDR_SUCCESS;
}
return( return_code );
}/* end of artaxvdr */

```



PRO*C Concurrent Programs
Concurrent Processing PRO*C Utility Routines
(Oracle Applications Object Library Reference Manual)

Tax Extension Error Reporting

Your Oracle Receivables application allows your Tax Vendor to report errors back to the calling program or form through a special function, `artaxerm()`. This function supports hard coded messages as well as

standard error messages from the Application Object Library Message Dictionary.

If you have a hard coded error message to report, you would call `artaxerm()` with the following parameters:

```
artaxerm(ARTAXERM_EXPANDED_TEXT,  
         (dvoid *)arg_struct,  
         "Expanded Text",  
         (sb4)0,  
         (text *)NULL );
```

If you plan to use a standard Application Object Library error message, you would call `artaxerm()` with the following parameters:

```
artaxerm(ARTAXERM-NLS_1_PARAM,  
         (dvoid *)arg_struct,  
         "Message Name",  
         (sb4)0,  
         (text *)token_value);
```

Tax Vendor Audit Requirements

Your Tax Vendor may maintain audit files to allow for automated tax filing and Government reporting. Before using the automated tax return feature, you should reconcile your Vendor's audit file with your Oracle Receivables application tax distributions for the same period.



Warning: The Tax Extension is not called from the Credit Memo or Adjustment forms. Therefore, any credit memo or adjustment transactions will not update your Vendor's tax audit file.

Database Access During a Call to the Tax Extension

If the Tax Extension is called from within the Enter Invoices form, you will have to go back to the database to fetch jurisdiction code and product code data. This is because the Tax Extension will be called before the form has posted each new line to the database. When fetching this jurisdiction data, you must use attributes held at the invoice header-level.

Commonly Asked Questions

Why would I want to use a Tax Vendor?

Within the United States of America, Sales and Use tax rates can be approximated to be a direct function of the Ship To address. This is the approach that Oracle takes, and is appropriate for many customers.

Larger customers, and other countries, may require tax calculation that takes into account one or more of the following parameters:

- Ship From
- Ship To
- Bill To
- Freight on Board (FOB)
- Quantity

These customers may also require regular monthly updates that reflect changing taxing regulations and rates.

When is the Tax Extension called to calculate a rate?

The Tax Extension will be called when Oracle Receivables or Oracle Order Entry calculates a sales or vat tax rate. The following concurrent programs and forms all do this:

- Oracle Order Entry Form
- Oracle Order Entry Credit Checking Form
- Oracle Order Entry Sales Acknowledgement Report
- Oracle Receivables AutoInvoice
- Oracle Receivables Invoice Entry
- Oracle Receivables Recurring Invoices

The Tax Extension is not supported from the following forms

- Oracle Receivables Quick Invoice Entry Form
- Oracle Receivables Enter Credit Memos Form
- Oracle Receivables Enter Adjustments Form

What Sales Tax Location flexfield structure should I select when implementing the Tax Extension within the U.S.?

Oracle Applications provides two possible default location structures for sites within the U.S.

- State.County.City
- State.City

The State.County.City structure provides the best accuracy in locating a tax jurisdiction and tax rate given a customer address. Additionally, the SQL*Loader control files, aravp.ctl and arvertex.ctl, support the State.County.City location flexfield structure. If you use the State.City structure, you will have to modify these SQL*Loader files to import your tax location and rate information.

How can I use the Tax Extension outside the U.S.?

The Tax Extension consists of a set of databases views and an interface to external C functions. You can control the tax basis and the number of tax codes per invoice line by simply customizing the database views. Any procedural logic can be implemented by C functions.

If I am implementing the Tax Extension outside the U.S., what Sales Tax Location Flexfield structure should I select?

The Sales Tax Location flexfield structure controls which segments of a customer address are required and validated, as defined by your 'Address Validation' option in the Define Systems Options form. The appropriate location structure will depend on upon your country and taxation system. If you do not require customer address validation or elect to implement flexible address formats, you should implement the structure, No Validation – Country. This structure will not prevent you from implementing the Tax Vendor Extension to calculate tax for your orders and invoices.

As the Tax Extension is not called from the Enter Credit Memo or Enter Adjustment forms, can I still use these transactions within Oracle Applications?

Yes, both of these transactions do support tax, but do not require a tax rate to be calculated for them. This is why the Tax Extension is not called.

The Enter Credit Memos form will copy over tax lines from the original invoice, reversing out the original tax accounting.

The Enter Adjustments form allows the operator to enter positive or negative tax adjustments, independent of sales tax rates.

Can I use my Tax Vendor to generate signature ready sales tax returns?

No. Since Credit Memos and Manual Adjustments are not posted to the Tax Vendor, these transactions are not included in the audit for sales tax. For this reason we recommend against enabling audit from within your implementation of the Tax Extension.



Suggestion: It may be possible to integrate the output of the US Sales Tax Report with Tax Vendor's software but you will need to contact your Tax Vendor to inquire about this.

What skills would I need to implement the Tax Extension?

Several skills would be required:

- A business analyst familiar with the Tax Department, Order Entry and Receivables.
- A C developer for the modification of the Tax Extension.
- Database Administrator skills for the customization of database views.

How do we make orders or invoices exempt from tax when using a Tax Vendor?

Oracle Receivables and Oracle Order Entry will automatically find customer exemptions based off the bill-to customer and ship-to site. If found, the exemption certificate number and reason will be passed down to the Tax Vendor.

How do we manage more complex customer exemptions, such as Customer, Inventory Item, and Site exemptions?

You can define these Customer Exemptions using setup programs supplied by your tax vendor. Once the exemption is entered within your tax vendor your implementation of the tax vendor extension need only ensure that the identifying information (Customer, Inventory Item, or Site) has been passed to the program for rate calculation.

How do we make products exempt for orders and invoices?

You can pass any segment of the Inventory Item Flexfield or any other attributes of the Inventory item to your Tax Vendor. Most Vendors allow you to control which products are exempt depending upon State and other order or invoice information.

How do we distinguish between tax rates calculated by Oracle and tax rates calculated by an installed Tax Vendor?

Oracle Receivables will mark tax lines for an invoice that have been calculated by an installed Tax Vendor.

Can the Oracle Order Entry form call a Tax Vendor?

Yes, Oracle Order Entry and Oracle Receivables share the same tax engine. This one engine will call a Tax Vendor if it is installed.

Does Oracle Order Entry store tax amounts in the database?

No, tax on an order is for information only and will be recalculated at the time of invoice creation. This is necessary because tax rates change over time and there could be a large gap between when an order is taken and when the customer is invoiced for the items on the order.

Can I use any columns in the tax views for customizations?

No, you should not. But, if you do need to redefine columns of your tax views, by ensuring that you only redefine the following columns, you will limit the impact of your customizations for future releases of Oracle Financials.

```
text  poo_code[31];          /* POO_ADDRESS_CODE */
sb2    i_poo;
text  poa_code[31];          /* POA_ADDRESS_CODE */
sb2    i_poa;
```

```

text  ship_from_code[31]; /* SHIP_FROM_ADDRESS_CODE */
sb2   i_ship_from;
text  ship_to_code[31];  /* SHIP_TO_ADDRESS_CODE */
sb2   i_ship_to;
text  fob[31];           /* FOB_CODE */
sb2   i_fob;
text  part_no[41];       /*PART_NUMBER or PRODUCT_CODE*/
sb2   i_part_no;

```

Can I integrate an automatic tax reporting system with Oracle Applications?

Yes, but the tax reporting system would have to use the output from the US Sales Tax Report when generating signature ready sales tax returns.

Within the U.S., how can I uniquely identify a taxing authority not defined by city limits?

You should implement the Sales Tax Location Flexfield structure, State.County.City, to validate customer addresses during data entry. By combining this structure with the five digit zip code for the customer address, Oracle Applications can accurately identify all but 160 recognized taxing authorities within the U.S.

If your site has customers within the 160 exception authorities, you can either enhance the five digit zip code to a nine digit zip code for these specific customers, or implement an additional descriptive flexfield for the specific jurisdiction code for each customer address.

Converting to a nine digit zip code would require:

- 1) Updating the context field of the user descriptive flexfield on the table, ar_location_rates, to be enterable.
- 2) Updating the customer address to store a nine digit zip code.
- 3) Updating the postal code ranges, tax rate assignments and jurisdiction code assignments to support a nine digit zip code.

How do I implement tax on freight?

Oracle Receivables will calculate tax on freight lines if you can enter freight as a revenue line item. Additionally, you can automatically present freight lines as revenue lines during the ship confirm process within Oracle Order Entry. If the Oracle Order Entry user profile, "Tax: Invoice Freight as Revenue" is set to "Yes", the Oracle Order Entry/Receivables Interface program will generate a standard invoice line for the freight amount, optionally using the inventory item defined by the user profile, "Tax: Inventory Item for Freight".

Is there a turnkey solution available to integrate external tax calculation programs with Oracle Applications?

Yes, please contact the Oracle Design and Migration Services Group for information on software, including documentation, which can be installed on Oracle Applications:

Design and Migration Services
500 Oracle Parkway MS659509
Redwood Shores, CA 94065

Index

A

ACE interface

- FA_ACE_BOOKS table, 2 – 54
- loading ACE table, 2 – 56
- manual loading, 2 – 59
- overview, 2 – 53
- update process, 2 – 54

AutoClear

- AP_RECON_DETAILS table, 7 – 8
- AP_RECON_SUMMARIES table, 7 – 9
- definitions, 7 – 5
- major features, 7 – 3
- overview, 7 – 2
- reconciling payment information, 7 – 17
- setting up, 7 – 10
- tables, 7 – 7
- using, 7 – 6

AutoInvoice

- account assignments, 10 – 31
- accounting rules, 6 – 18
- credit memos, 10 – 43
- customer bank accounts, 10 – 22
- deriving dates, 10 – 45
- finance charges, 10 – 31
- freight lines, 10 – 23
- grouping rules, 10 – 38
- integrating Order Entry with Receivables, 6 – 15
- invoices, 10 – 42
- invoicing rules, 6 – 18
- line ordering rules, 10 – 41
- overview, 10 – 2

payment methods, 10 – 22

preparations, 10 – 9

RA_INTERFACE_DIST table, 10 – 102

RA_INTERFACE_LINES table, 6 – 24,
10 – 51

RA_INTERFACE_SALESCREDIT table,
6 – 45

RA_INTERFACE_SALESCREDITS table,
10 – 98

setup data, 10 – 15

tax, 10 – 24

transaction flexfields, 10 – 33

transferring data, 10 – 18

using, 10 – 20

validation, 10 – 19

AutoLockbox

AR_PAYMENTS_INTERFACE table,
10 – 122

importing data, 10 – 116, 10 – 122

overview, 10 – 109

preparations, 10 – 116

questions, 10 – 140

reviewing receipts, 10 – 121

updating balance, 10 – 121

validation, 10 – 117

B

Budget interface

entering information, 2 – 41

FA_BUDGET_INTERFACE table, 2 – 42

overview, 2 – 38

prerequisites, 2 – 39

SQL Loader script, 2 – 43

Budget upload

- GL_BUDGET_INTERFACE table, 4 – 36
- overview, 4 – 33
- prerequisites, 4 – 35
- running, 4 – 45
- status report, 4 – 45
- transferring data, 4 – 44
- uploading budget information, 4 – 42

C

Creating assets from invoices. *See* Mass Additions

Customer interface

- generating original system references, 10 – 197
- importing data, 10 – 150
- overview, 10 – 143
- preparations, 10 – 149
- RA_CONTACT_PHONES_INTERFACE table, 10 – 181
- RA_CUST_PAY_METHOD_INTERFACE table, 10 – 192
- RA_CUSTOMER_BANKS_INTERFACE table, 10 – 185
- RA_CUSTOMER_PROFILES_INTERFACE table, 10 – 167
- RA_CUSTOMERS_INTERFACE table, 10 – 158
- relationships (valid), 10 – 195
- using, 10 – 156, 10 – 157

D

Demand interface

- ATP check, 5 – 12
- ATS query, 5 – 12
- ATS Quick Pick, 5 – 12
- demand add, 5 – 7
- demand and reservation add, 5 – 10
- function descriptions, 5 – 6
- MTL_DEMAND_INTERFACE table, 5 – 14
- overview, 5 – 2
- pick release, 5 – 11

- reservation add, 5 – 9
- reservation modify, 5 – 10
- resolving failed rows, 5 – 21
- setting up, 5 – 4
- update forecast attributes, 5 – 10
- validation, 5 – 20

I

Importing orders. *See* OrderImport

Importing transactions. *See* Transaction Import

Interface tables

- AP_EXPENSE_REPORT_HEADERS, 7 – 25
- AP_EXPENSE_REPORT_LINES, 7 – 37
- AP_INVOICE_DISTRIBUTIONS, 7 – 51
- AP_INVOICES, 7 – 51
- AP_RECON_DETAILS, 7 – 8
- AP_RECON_SUMMARIES, 7 – 9
- AR_PAYMENTS_INTERFACE, 10 – 122
- FA_ACE_BOOKS, 2 – 54
- FA_BUDGET_INTERFACE, 2 – 42
- FA_MASS_ADDITONS, 2 – 20
- FA_PRODUCTION_INTERFACE, 2 – 50
- GL_BUDGET_INTERFACE, 4 – 36
- GL_INTERFACE, 4 – 9
- MTL_DEMAND_INTERFACE, 5 – 14
- MTL_REPLENISH_HEADERS_INT, 5 – 49
- MTL_SERIAL_NUMBERS_INTERFACE, 5 – 42
- MTL_TRANSACTION_LOTS_INTERFACE, 5 – 41
- MTL_TRANSACTIONS_INTERFACE, 5 – 28
- PA_TRANSACTION_INTERFACE, 8 – 7
- PO_DISTRIBUTIONS, 7 – 59
- PO_HEADERS, 7 – 52
- PO_LINE_LOCATIONS, 7 – 56
- PO_LINES, 7 – 55
- PO_RELEASES, 7 – 54
- PO_REQUISITIONS_INTERFACE, 9 – 6
- RA_CONTACT_PHONES_INTERFACE, 10 – 181
- RA_CUST_PAY_METHOD_INTERFACE, 10 – 192

- RA_CUSTOMER_BANKS_INTERFACE, 10 – 185
- RA_CUSTOMER_PROFILES_INTERFACE, 10 – 167
- RA_CUSTOMERS_INTERFACE, 10 – 158
- RA_INTERFACE_DIST, 10 – 102
- RA_INTERFACE_LINES, 6 – 24, 10 – 51
- RA_INTERFACE_SALESCREDITS, 10 – 98
- SO_HEADERS_INTERFACE, 6 – 72
- SO_LINE_DETAILS_INTERFACE, 6 – 118
- SO_LINES_INTERFACE, 6 – 99
- SO_PRICE_ADJUSTMENTS_INTERFACE, 6 – 124
- SO_SALES_CREDITS_INTERFACE, 6 – 130

Internal Sales Orders, not interfaced to Receivables, 6 – 20

Invoice import

- AP_EXPENSE_REPORT_HEADERS table, 7 – 25

- AP_EXPENSE_REPORT_LINES table, 7 – 37

- definitions, 7 – 23

- major features, 7 – 21

- overview, 7 – 20

- setting up, 7 – 46

Invoice information. *See* AutoInvoice

Invoices, importing from other systems, 7 – 20

J

Journal import

- budget data, 4 – 23

- encumbrance data, 4 – 23

- GL_INTERFACE table, 4 – 9

- importing data, 4 – 7

- importing from feeder system, 1 – 4

- importing specialized data, 4 – 22

- intercompany data, 4 – 23

- major features, 4 – 3

- multi-currency data, 4 – 22

- overview, 4 – 2

- preparing Oracle General Ledger, 4 – 26

- prerequisites, 4 – 7

- specifying Accounting Flexfields, 4 – 11

- statistical data, 4 – 23

- using, 4 – 28

- validation, 4 – 24

M

Mass additions

- defining, 2 – 10

- FA_MASS_ADDITIONS table, 2 – 20

- finishing import, 2 – 35

- importing information, 2 – 19

- importing steps, 2 – 5

- loading data, 2 – 16

- overview, 2 – 2

- planning import, 2 – 7

Matching. *See* Purchase orders

Multiple Organization Support

- AutoInvoice, 10 – 69, 10 – 100, 10 – 105, 10 – 108, 10 – 132

- importing invoices, 7 – 34, 7 – 42

O

Open demand interface. *See* Demand interface

Open interfaces

- importing information into Oracle Financials, 1 – 2

- overview, 1 – 1

Open replenishment interface. *See* replenishment interface

Open requisitions

- derived data, 9 – 17

- optional data, 9 – 18

- overview, 9 – 3

- PO_REQUISITONS_INTERFACE table, 9 – 6

- required data, 9 – 16

- resolving failed rows, 9 – 20

- setup, 9 – 5

- validation, 9 – 19

Open transaction interface. *See* Transaction interface

- Oracle Assets
 - ACE interface, 2 – 53
 - budget interface, 2 – 38
 - converting asset information, 2 – 2
 - mass additions interface, 2 – 2
 - production interface, 2 – 46
- Oracle General Ledger
 - budget upload, 4 – 33
 - journal import, 4 – 2
- Oracle Inventory
 - Open demand interface, 5 – 22
 - open demand interface, 5 – 2
 - open item interface, 5 – 58
 - Open replenishment interface, 5 – 46
- Oracle Order Entry
 - interfacing with Receivables, 6 – 15
 - OrderImport, 6 – 48
- Oracle Order Entry / Shipping, Ship Confirm
 - Open Interface, 6 – 2
- Oracle Payables
 - integrating with Oracle Purchasing, 9 – 23
 - purge, 9 – 23
- Oracle Payables Applications
 - AutoClear, 7 – 2
 - integrating purchase orders, 7 – 50
 - Invoice import, 7 – 20
- Oracle Purchasing
 - integrating with Oracle Payables, 9 – 23
 - open requisitions interface, 9 – 2
 - purge, 9 – 23
- Oracle Receivables
 - interfacing with Order Entry, 6 – 15
 - sales interface, 10 – 198
- Oracle Receivables Applications
 - AutoInvoice, 10 – 2
 - AutoLockbox, 10 – 109
 - customer interface, 10 – 143
 - Sales Tax Rate Interface, 10 – 198
 - Tax Vendor Extension, 10 – 219
- OrderImport
 - changes to imported orders, 6 – 61
 - complete orders, 6 – 49, 6 – 70
 - configurations, 6 – 59
 - credit checking, 6 – 53
 - defaulting, 6 – 53, 6 – 68, 6 – 70
 - entered state, 6 – 49
 - holds, 6 – 53
 - importing data, 6 – 57
 - internal sales orders, 6 – 53
 - items, 6 – 58
 - not null columns, 6 – 59
 - notes, 6 – 53
 - operation code, 6 – 61
 - order scheduling, 6 – 50
 - overview, 6 – 48
 - prerequisites, 6 – 55
 - price adjustments, 6 – 50, 6 – 51, 6 – 71
 - pricing, 6 – 50, 6 – 51
 - reporting, 6 – 54
 - sales credits, 6 – 52, 6 – 70
 - schedule details, 6 – 50, 6 – 60, 6 – 71
 - security rules, 6 – 53, 6 – 70
 - ship sets, 6 – 61
 - shipment schedules, 6 – 60
 - shipping complete orders, 6 – 61
 - standard values, 6 – 53, 6 – 68, 6 – 70
 - using IDs, 6 – 66
 - using multi-currency, 6 – 61
 - validation, 6 – 68 to 6 – 71
- Other Information Sources, vi

P

- Production interface
 - entering information, 2 – 49
 - FA_PRODUCTION_INTERFACE table, 2 – 50
 - overview, 2 – 46
 - SQL Loader script, 2 – 51
- Purchase orders
 - AP_INVOICE_DISTRIBUTIONS table, 7 – 51
 - AP_INVOICES table, 7 – 51
 - encumbrance accounting, 7 – 68
 - integrating with Oracle Payables, 7 – 50
 - matching, 7 – 63
 - overview, 7 – 50

- PO_DISTRIBUTIONS table, 7 – 59
- PO_HEADERS table, 7 – 52
- PO_LINE_LOCATIONS, 7 – 56
- PO_LINES table, 7 – 55
- PO_RELEASES table, 7 – 54
- purging information, 7 – 69
- understanding matching tables, 7 – 50
- using AutoApproval, 7 – 66

Purge

- overview, 9 – 23
- tables, 9 – 23

R

Replenishment interface

- fixing failed transactions, 5 – 57
- MTL_REPLENISH_HEADERS_INT table, 5 – 49
- overview, 5 – 46
- setting up, 5 – 48
- validation, 5 – 55
- viewing failed transactions, 5 – 56

ReqImport, rescheduling requisitions, 9 – 21

S

Sales interface

- action codes, 10 – 203
- city names, 10 – 214
- date adjustments, 10 – 209
- overview, 10 – 198
- preparations, 10 – 200
- program updates, 10 – 214
- report termination, 10 – 215
- SQL Loader script, 10 – 215, 10 – 216
- transferring data, 10 – 202

Ship Confirm Interface Tables,

- SHP_PICKING_HEADERS_INTERFACE, 6 – 5, 6 – 8

Ship Confirm Interface tables,

- SHP_PICKING_HEADERS_INTERFACE, 6 – 11

Ship Confirm Open Interface

See also replenishment interface

- fixing failed transactions, 6 – 13
- functional overview, 6 – 2

Inserting into Shipping Tables, 6 – 5

Shipping Transaction Manager, 6 – 13

- SHP_PICKING_HEADERS_INTERFACE table, 6 – 5

SHP_PICKING_LINE_DETAILS

- INTERFACE_V table, 6 – 8, 6 – 11

- transactions, 6 – 3

- validation, 6 – 12

- viewing failed transactions, 6 – 13

Shipping Transaction Manager, 6 – 13

Spreadsheet interface

- budget upload, 4 – 33

- entering data, 4 – 38

- interface table, 4 – 36

- major features, 4 – 33

- reporting data, 4 – 47

- reviewing status report, 4 – 45

- software requirements, 4 – 43

- spreadsheet contents, 4 – 37

- transferring data, 4 – 44

- using, 4 – 33

- validating data, 4 – 45

T

Tax. *See* sales interface

Tax Vendor Hook, overview, 10 – 219

Transaction Import, 8 – 2

- correcting rejected transactions, 8 – 18

- interface control table, 8 – 6

- interface table, 8 – 4

- not null columns, 8 – 5

- null columns, 8 – 5

- overview, 8 – 2

- populating interface table, 8 – 14

- purging transactions, 8 – 14

- resolving exceptions, 8 – 16

- running, 8 – 15

- system assigned columns, 8 – 6

- table descriptions, 8 – 6

transaction sources, 8 – 14	overview, 5 – 22
validation, 8 – 2 to 8 – 4	resolving failed rows, 5 – 44
viewing successful transactions, 8 – 16	setting up, 5 – 26
Transaction interface	validation, 5 – 44
MTL_SERIAL_NUMBERS_INTERFACE	
table, 5 – 42	
MTL_TRANSACTION_LOTS_INTERFACE	
table, 5 – 41	
MTL_TRANSACTIONS_INTERFACE table,	Transactions
5 – 28	importing, 8 – 2
	PA_TRANSACTION_INTERFACE table,
	8 – 7

Reader's Comment Form

Oracle Financials and Oracle Government Financials Open Interfaces Manual A36128–2

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information we use for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual? What did you like least about it?

If you find any errors or have any other suggestions for improvement, please indicate the topic, chapter, and page number below:

Please send your comments to:

Oracle Applications Documentation Manager
Oracle Corporation
500 Oracle Parkway
Redwood Shores, CA 94065 USA
Phone: (415) 506–7000 Fax: (415) 506–7200

If you would like a reply, please give your name, address, and telephone number below:

Thank you for helping us improve our documentation.