

Oracle® Receivables
Reference Guide
Release 12.1
Part No. E13512-02

April 2009

Copyright © 1988, 2009, Oracle and/or its affiliates. All rights reserved.

Primary Author: Mathew Daniel

Contributing Author: Charles Ahern, Stephen R. Damiani, Melanie Heisler, Essan Ni Jirman, Shivranjini Krishnamurthy, Robert MacIsaac, Kristin Penaskovic, Vijay Tiwary

Contributor: Rohit Kathuria, Mitesh Kumbhat, Amarnath Molugu, Prabhat Patel

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

This software and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third party content, products and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third party content, products or services.

Contents

Send Us Your Comments

Preface

1 Using Oracle Receivables APIs

Major Features.....	1-1
Solution Outline.....	1-2

2 Adjustment API User Notes

Overview.....	2-1
API Usage.....	2-1
Ar_Adjust_pub.Create_Adjustment.....	2-4
Ar_Adjust_pub.Approve_Adjustment.....	2-13
Ar_Adjust_pub.Modify_Adjustment.....	2-18
Ar_Adjust_pub.Reverse_Adjustment.....	2-23
Ar_Adjust_pub.Create_Linelevel_Adjustment.....	2-25
Messages.....	2-28

3 Credit Memo Approval and Creation API User Notes

Overview.....	3-1
API Usage.....	3-2
AR_CREDIT_MEMO_API_PUB.Create_Request.....	3-2
AR_CREDIT_MEMO_API_PUB.Get_Request_Status.....	3-9
AR_CM_API_PUB.Apply_On_Account.....	3-13
AR_CM_API_PUB.Unapply_On_Account.....	3-13
Messages.....	3-13

4 Credit Memo Application API User Notes

Overview.....	4-1
API Usage.....	4-1
ar_cm_application_pub.activity_application.....	4-1
ar_cm_application_pub.activity_unapplication.....	4-11
Messages.....	4-15

5 Deposit API User Notes

Overview.....	5-1
API Usage.....	5-1
AR_DEPOSIT_API_PUB.Create_deposit.....	5-2
AR_DEPOSIT_API_PUB.insert_non_rev_salescredit.....	5-27
Messages.....	5-32

6 Invoice Creation API User Notes

Overview.....	6-1
API Usage.....	6-2
AR_INVOICE_API_PUB.....	6-2

7 Prepayments API User Notes

Overview.....	7-1
API Usage.....	7-2
AR_PREPAYMENTS_PUB.Create_Prepayment.....	7-2
AR_PREPAYMENTS_PUB.Get_Installment.....	7-9
Messages.....	7-11

8 Receipt API User Notes

Overview.....	8-1
API Usage.....	8-2
Ar_receipt_api_pub.Create_cash.....	8-3
Ar_receipt_api_pub.Apply.....	8-20
Ar_receipt_api_pub.Create_and_apply.....	8-34
Ar_receipt_api_pub.Unapply.....	8-54
Ar_receipt_api_pub.Apply_on_account.....	8-60
Ar_receipt_api_pub.Unapply_on_account.....	8-65
Ar_receipt_api_pub.Reverse.....	8-69
Ar_receipt_api_pub.activity_application.....	8-75
Ar_receipt_api_pub.activity_unapplication.....	8-83

Ar_receipt_api_pub.Create_misc.....	8-87
Ar_receipt_api_pub.apply_other_account.....	8-101
Ar_receipt_api_pub.unapply_other_account.....	8-108
Ar_receipt_api_pub.apply_open_receipt.....	8-112
Ar_receipt_api_pub.unapply_open_receipt.....	8-119
Ar_receipt_api_pub.Create_apply_on_acc.....	8-121
Messages.....	8-137

9 Revenue Adjustment API User Notes

Overview.....	9-1
API Usage.....	9-2
AR_RevenueAdjust_PUB.Uearn_Revenue.....	9-2
AR_RevenueAdjust_PUB.Earn_Revenue.....	9-16
AR_RevenueAdjust_PUB.Transfer_Sales_Credits.....	9-18
AR_RevenueAdjust_PUB.Add_Non_Revenue_Sales_Credits.....	9-23
AR_Revenueadjust_PUB.Record_Customer_Acceptance.....	9-27
AR_Revenueadjust_PUB.Update_Contingency_Expirations.....	9-28
Messages.....	9-31

A Predefined Setup for Oracle Subledger Accounting

Data that Oracle Receivables Predefines for Oracle Subledger Accounting.....	A-1
--	-----

B Oracle Receivables Table and Column Descriptions

AutoInvoice Table and Column Descriptions.....	B-1
Lockbox Table and Column Descriptions.....	B-62
Receipt and QuickCash Tables.....	B-62
Lockbox Interface Table and Column Descriptions.....	B-62

C Seeded Match Rules

Seeded Search Match Rules.....	C-1
SAMPLE: BASIC SEARCH RULE.....	C-1
SAMPLE: ADVANCED SEARCH RULE.....	C-4

D XML Transactions

XML Receivables Documents Mapping.....	D-1
Process Invoice XML Message Map.....	D-1
Confirm BOD Message Map.....	D-5
Transaction Limitations.....	D-6

Index

Send Us Your Comments

Oracle Receivables Reference Guide, Release 12.1

Part No. E13512-02

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document. Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

Note: Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the new Applications Release Online Documentation CD available on My Oracle Support and www.oracle.com. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: appsdoc_us@oracle.com

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at www.oracle.com.

Preface

Intended Audience

Welcome to Release 12.1 of the *Oracle Receivables Reference Guide*.

This guide assumes you have a working knowledge of the following:

- The principles and customary practices of your business area.
- Computer desktop application usage and terminology

If you have never used Oracle Applications, we suggest you attend one or more of the Oracle Applications training classes available through Oracle University.

See Related Information Sources on page x for more Oracle Applications product information.

TTY Relay Access to Oracle Support Services

To reach AT&T Customer Assistants, dial 711 or 1.800.855.2880. An AT&T Customer Assistant will relay information between the customer and Oracle Support Services at 1.800.223.1711. Complete instructions for using the AT&T relay services are available at <http://www.consumer.att.com/relay/tty/standard2.html>. After the AT&T Customer Assistant contacts Oracle Support Services, an Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by

the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at <http://www.oracle.com/accessibility/>.

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

Structure

- 1 Using Oracle Receivables APIs**
- 2 Adjustment API User Notes**
- 3 Credit Memo Approval and Creation API User Notes**
- 4 Credit Memo Application API User Notes**
- 5 Deposit API User Notes**
- 6 Invoice Creation API User Notes**
- 7 Prepayments API User Notes**
- 8 Receipt API User Notes**
- 9 Revenue Adjustment API User Notes**
- A Predefined Setup for Oracle Subledger Accounting**
- B Oracle Receivables Table and Column Descriptions**
- C Seeded Match Rules**
- D XML Transactions**

Related Information Sources

This document is included on the Oracle Applications Document Library, which is supplied in the Release 12 DVD Pack. You can download soft-copy documentation as PDF files from the Oracle Technology Network at <http://otn.oracle.com/documentation>, or you can purchase hard-copy documentation from the Oracle Store at <http://oraclestore.oracle.com>. The Oracle E-Business Suite Documentation Library Release 12 contains the latest information, including any documents that have changed significantly between releases. If substantial changes to this book are necessary, a revised version will be made available on the online documentation CD on My Oracle Support.

If this guide refers you to other Oracle Applications documentation, use only the Release 12 versions of those guides.

For a full list of documentation resources for Oracle Applications Release 12, see Oracle Applications Documentation Resources, Release 12, Document 394692.1 on My Oracle Support.

Online Documentation

All Oracle Applications documentation is available online (HTML or PDF).

- **PDF** - PDF documentation is available for download from the Oracle Technology Network at <http://otn.oracle.com/documentation>.
- **Online Help** - Online help patches (HTML) are available on My Oracle Support.
- **My Oracle Support** - My Oracle Support lets you browse the knowledge base, from a single product page, to find all documents for that product area. Use My Oracle Support to search for release-specific information, such as FAQs, recent patches, alerts, white papers, troubleshooting tips, and other archived documents.
- **Oracle eBusiness Suite Electronic Technical Reference Manuals** - Each Electronic Technical Reference Manual (eTRM) contains database diagrams and a detailed description of database tables, forms, reports, and programs for a specific Oracle Applications product. This information helps you convert data from your existing applications and integrate Oracle Applications data with non-Oracle applications, and write custom reports for Oracle Applications products. Oracle eTRM is available on My Oracle Support.

Related Guides

You should have the following related books on hand. Depending on the requirements of your particular installation, you may also need additional manuals or guides.

Oracle Applications Installation Guide: Using Rapid Install:

This book is intended for use by anyone who is responsible for installing or upgrading Oracle Applications. It provides instructions for running Rapid Install either to carry out a fresh installation of Oracle Applications Release 12, or as part of an upgrade from Release 11i to Release 12. The book also describes the steps needed to install the technology stack components only, for the special situations where this is applicable.

Oracle Applications Upgrade Guide: Release 11i to Release 12:

This guide provides information for DBAs and Applications Specialists who are responsible for upgrading a Release 11i Oracle Applications system (techstack and products) to Release 12. In addition to information about applying the upgrade driver, it outlines pre-upgrade steps and post-upgrade steps, and provides descriptions of product-specific functional changes and suggestions for verifying the upgrade and reducing downtime.

Oracle Applications Patching Procedures:

This guide describes how to patch the Oracle Applications file system and database using AutoPatch, and how to use other patching-related tools like AD Merge Patch, OAM Patch Wizard, and OAM Registered Flagged Files. Describes patch types and structure, and outlines some of the most commonly used patching procedures. Part of Maintaining Oracle Applications, a 3-book set that also includes Oracle Applications Maintenance Utilities and Oracle Applications Maintenance Procedures.

Oracle Applications Maintenance Utilities:

This guide describes how to run utilities, such as AD Administration and AD Controller, used to maintain the Oracle Applications file system and database. Outlines the actions performed by these utilities, such as monitoring parallel processes, generating Applications files, and maintaining Applications database entities. Part of Maintaining Oracle Applications, a 3-book set that also includes Oracle Applications Patching Procedures and Oracle Applications Maintenance Procedures.

Oracle Applications Maintenance Procedures:

This guide describes how to use AD maintenance utilities to complete tasks such as compiling invalid objects, managing parallel processing jobs, and maintaining snapshot information. Part of Maintaining Oracle Applications, a 3-book set that also includes Oracle Applications Patching Procedures and Oracle Applications Maintenance Utilities.

Oracle Alert User's Guide:

This guide explains how to define periodic and event alerts to monitor the status of your Oracle Applications data.

Oracle Application Framework Developer's Guide:

This guide contains the coding standards followed by the Oracle Applications development staff to produce applications built with Oracle Application Framework. This guide is available in PDF format on My Oracle Support and as online documentation in JDeveloper 10g with Oracle Application Extension.

Oracle Application Framework Personalization Guide:

This guide covers the design-time and run-time aspects of personalizing applications built with Oracle Application Framework.

Oracle Applications Concepts:

This book is intended for all those planning to deploy Oracle E-Business Suite Release 12, or contemplating significant changes to a configuration. After describing the Oracle Applications architecture and technology stack, it focuses on strategic topics, giving a broad outline of the actions needed to achieve a particular goal, plus the installation and configuration choices that may be available.

Oracle Applications Flexfields Guide:

This guide 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 guide also provides information

on creating custom reports on flexfields data.

Oracle Applications Multiple Organizations Implementation Guide:

This guide describes the multiple organizations concepts in Oracle Applications. It describes in detail on setting up and working effectively with multiple organizations in Oracle Applications.

Oracle Application Server Adapter for Oracle Applications User's Guide:

This guide covers the use of OracleAS Adapter in developing integrations between Oracle applications and trading partners.

Please note that this guide is in the Oracle Application Server 10g Documentation Library.

Oracle Applications System Administrator's Guide Documentation Set:

This documentation set provides planning and reference information for the Oracle Applications System Administrator. *Oracle Applications System Administrator's Guide - Configuration* contains information on system configuration steps, including defining concurrent programs and managers, enabling Oracle Applications Manager features, and setting up printers and online help. *Oracle Applications System Administrator's Guide - Maintenance* provides information for frequent tasks such as monitoring your system with Oracle Applications Manager, administering Oracle E-Business Suite Secure Enterprise Search, managing concurrent managers and reports, using diagnostic utilities including logging, managing profile options, and using alerts. *Oracle Applications System Administrator's Guide - Security* describes User Management, data security, function security, auditing, and security configurations.

Oracle Applications User's Guide:

This guide explains how to navigate, enter data, query, and run reports using the user interface (UI) of Oracle Applications. This guide also includes information on setting user profiles, as well as running and reviewing concurrent requests.

Oracle E-Business Suite Diagnostics User's Guide:

This manual contains information on implementing, administering, and developing diagnostics tests in the Oracle E-Business Suite Diagnostics framework.

Oracle E-Business Suite Integrated SOA Gateway User's Guide:

This guide describes the high level service enablement process, explaining how users can browse and view the integration interface definitions and services residing in Oracle Integration Repository.

Oracle E-Business Suite Integrated SOA Gateway Developer's Guide:

This guide describes how system integration developers can perform end-to-end service integration activities. These include orchestrating discrete Web services into meaningful end-to-end business processes using business process execution language (BPEL), and deploying BPEL processes at run time.

It also explains in detail how to invoke Web services using the Service Invocation

Framework. This includes defining Web service invocation metadata, invoking Web services, managing errors, and testing the Web service invocation.

Oracle E-Business Suite Integrated SOA Gateway Implementation Guide:

This guide explains how integration repository administrators can manage and administer the service enablement process (based on the service-oriented architecture) for both native packaged public integration interfaces and composite services (BPEL type). It also describes how to invoke Web services from Oracle E-Business Suite by employing the Oracle Workflow Business Event System; how to manage Web service security; and how to monitor SOAP messages.

Oracle Workflow Administrator's Guide:

This guide explains how to complete the setup steps necessary for any product that includes workflow-enabled processes. It also describes how to manage workflow processes and business events using Oracle Applications Manager, how to monitor the progress of runtime workflow processes, and how to administer notifications sent to workflow users.

Oracle Workflow Developer's Guide:

This guide explains how to define new workflow business processes and customize existing Oracle Applications-embedded workflow processes. It also describes how to define and customize business events and event subscriptions.

Oracle Workflow User's Guide:

This guide describes how users can view and respond to workflow notifications and monitor the progress of their workflow processes.

Oracle Workflow API Reference:

This guide describes the APIs provided for developers and administrators to access Oracle Workflow.

Oracle Workflow Client Installation Guide:

This guide describes how to install the Oracle Workflow Builder and Oracle XML Gateway Message Designer client components for Oracle E-Business Suite.

Oracle Financials and Oracle Procurement Functional Upgrade Guide: Release 11i to Release 12:

This guide provides detailed information about the functional impacts of upgrading Oracle Financials and Oracle Procurement products from Release 11i to Release 12. This guide supplements the Oracle Applications Upgrade Guide: Release 11i to Release 12.

Oracle Financials Concepts Guide:

This guide describes the fundamental concepts of Oracle Financials. The guide is intended to introduce readers to the concepts used in the applications, and help them compare their real world business, organization, and processes to those used in the applications.

Oracle Financials Glossary:

The glossary includes definitions of common terms that are shared by all Oracle Financials products. In some cases, there may be different definitions of the same term for different Financials products. If you are unsure of the meaning of a term you see in an Oracle Financials guide, please refer to the glossary for clarification. You can find the glossary in the online help or in the *Oracle Financials Implementation Guide*.

Oracle Financials Implementation Guide:

This guide provides information on how to implement the Oracle Financials E-Business Suite. It guides you through setting up your organizations, including legal entities, and their accounting, using the Accounting Setup Manager. It covers intercompany accounting and sequencing of accounting entries, and it provides examples.

Oracle E-Business Tax User Guide:

This guide describes the entire process of setting up and maintaining tax configuration data, as well as applying tax data to the transaction line. It describes the entire regime-to-rate setup flow of tax regimes, taxes, statuses, rates, recovery rates, tax jurisdictions, and tax rules. It also describes setting up and maintaining tax reporting codes, fiscal classifications, tax profiles, tax registrations, configuration options, and third party service provider subscriptions. You also use this manual to maintain migrated tax data for use with E-Business Tax.

Oracle E-Business Tax Implementation Guide:

This guide provides a conceptual overview of the E-Business Tax tax engine, and describes the prerequisite implementation steps to complete in other applications in order to set up and use E-Business Tax. The guide also includes extensive examples of setting up country-specific tax requirements.

Oracle E-Business Tax Reporting Guide:

This guide explains how to run all tax reports that make use of the E-Business Tax data extract. This includes the Tax Reporting Ledger and other core tax reports, country-specific VAT reports, and Latin Tax Engine reports.

Oracle E-Business Tax: Vertex Q-Series and Taxware Sales/Use Tax System Implementation Guide:

This guide explains how to setup and use the services of third party tax service providers for US Sales and Use tax. The tax service providers are Vertex Q-Series and Taxware Sales/Use Tax System. When implemented, the Oracle E-Business Tax service subscription calls one of these tax service providers to return a tax rate or amount whenever US Sales and Use tax is calculated by the Oracle E-Business Tax tax engine. This guide provides setup steps, information about day-to-day business processes, and a technical reference section.

Oracle General Ledger Implementation Guide:

This guide provides information on how to implement Oracle General Ledger. Use this guide to understand the implementation steps required for application use, including how to set up Accounting Flexfields, Accounts, and Calendars.

Oracle General Ledger Reference Guide:

This guide provides detailed information about setting up General Ledger Profile Options and Applications Desktop Integrator (ADI) Profile Options.

Oracle General Ledger User's Guide:

This guide provides information on how to use Oracle General Ledger. Use this guide to learn how to create and maintain ledgers, ledger currencies, budgets, and journal entries. This guide also includes information about running financial reports.

Oracle iReceivables Implementation Guide:

This guide provides information on how to implement Oracle iReceivables. Use this guide to understand the implementation steps required for application use, including how to set up and configure iReceivables, and how to set up the Credit Memo Request workflow. There is also a chapter that provides an overview of major features available in iReceivables.

Oracle Order Management Documentation Set:

Use the *Oracle Order Management User's Guide* and *Oracle Order Management Implementation Manual* to learn about credit checking and credit usage rule sets.

Oracle Public Sector Financials User Guide:

This guide describes how to set up and administer Oracle Public Sector Advanced Features. It describes Encumbrance Reconciliation Reports, GASB 34/35 Asset Accounting, and Funds Available Enhancements.

Oracle Receivables Implementation Guide:

This guide provides you with information on how to implement Oracle Receivables. Use this guide to understand the implementation steps required for application use, including how to set up customers, transactions, receipts, accounting, tax, and collections. This guide also includes a comprehensive list of profile options that you can set to customize application behavior.

Oracle Receivables User Guide:

This guide provides you with information on how to use Oracle Receivables. Use this guide to learn how to create and maintain transactions and bills receivable, enter and apply receipts, enter customer information, and manage revenue. This guide also includes information about accounting in Receivables. Use the Standard Navigation Paths appendix to find out how to access each Receivables window.

Oracle Subledger Accounting Implementation Guide:

This guide provides setup information for Oracle Subledger Accounting features, including the Accounting Methods Builder. You can use the Accounting Methods Builder to create and modify the setup for subledger journal lines and application accounting definitions for Oracle subledger applications. This guide also discusses the reports available in Oracle Subledger Accounting and describes how to inquire on subledger journal entries.

Oracle Trading Community Architecture User Guide:

This guide describes the Oracle Trading Community Architecture (TCA) and how to use features from the Trading Community Manager responsibility to create, update, enrich, and cleanse the data in the TCA Registry. It also describes how to use Resource Manager to define and manage resources.

Oracle Trading Community Architecture Administration Guide:

This guide describes how to administer and implement Oracle Trading Community Architecture (TCA). You set up, control, and manage functionality that affects data in the TCA Registry. It also describes how to set up and use Resource Manager to manage resources.

Oracle Trading Community Architecture Reference Guide:

This guide contains seeded relationship types, seeded Data Quality Management data, D and B data elements, Bulk Import interface table fields and validations, and a comprehensive glossary. This guide supplements the documentation for Oracle Trading Community Architecture and all products in the Oracle Customer Data Management family.

Oracle Trading Community Architecture Technical Implementation Guide:

This guide explains how to use the public Oracle Trading Community Architecture application programming interfaces (APIs) and develop callouts based on Oracle Workflow Business Events System (BES). For each API, this guide provides a description of the API, the PL/SQL procedure, and the Java method, as well as a table of the parameter descriptions and validations. For each BES callout, this guide provides the name of the logical entity, its description, and the ID parameter name. Also included are setup instructions and sample code.

Integration Repository

The Oracle Integration Repository is a compilation of information about the service endpoints exposed by the Oracle E-Business Suite of applications. It provides a complete catalog of Oracle E-Business Suite's business service interfaces. The tool lets users easily discover and deploy the appropriate business service interface for integration with any system, application, or business partner.

The Oracle Integration Repository is shipped as part of the E-Business Suite. As your instance is patched, the repository is automatically updated with content appropriate for the precise revisions of interfaces in your environment.

Do Not Use Database Tools to Modify Oracle Applications Data

Oracle STRONGLY RECOMMENDS that you never use SQL*Plus, Oracle Data Browser, database triggers, or any other tool to modify Oracle Applications data unless otherwise instructed.

Oracle provides powerful tools you can use to create, store, change, retrieve, and

maintain information in an Oracle database. But if you use Oracle tools such as SQL*Plus 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, 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 to modify your data, Oracle Applications automatically checks that your changes are valid. Oracle Applications also keeps track of who changes information. 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.

Using Oracle Receivables APIs

Major Features

Before you begin....

Initialization of ARP_STANDARD and ARP_GLOBAL

Custom code that uses AR or HZ APIs will set the ORG_ID via `dbms_application_info.set_client_info()` and then call the APIs. The APIs in turn might access either ARP_STANDARD and ARP_GLOBAL, which initialize the global variables that are used across Oracle Receivables when the package is first called. Most of these global variable values are organization dependent, and the first such call sets the global variables based on the current ORG_ID.

If additional custom code then changes the ORG_ID via another call to `dbms_application_info.set_client_info()`, then the ORG context changes, *but the ARP_STANDARD and ARP_GLOBAL context does not.*

In such cases, you should explicitly re-initialize the global variables by a call to these two public procedures:

1. ARP_GLOBAL.INIT_GLOBAL: For setting public variables in ARP_GLOBAL.
2. ARP_STANDARD.INIT_STANDARD: For setting public variables in ARP_STANDARD.

Flexibility

Per Oracle API coding standards, the various Oracle Receivables APIs let you specify an ID or its associated value for any attribute that is an INPUT parameter of the API.

If both an ID and value have been specified, then the ID takes precedence over the value. This provides a wide degree of flexibility when using the API, both as a base table of the form and as a server-side routine call from the PL/SQL code.

The extensive defaulting mechanism for the input parameters ensures that you will be able to achieve your basic business needs by calling the relevant APIs with a minimum number of parameters. This gives you many options to achieve your requirements when you call the relevant API.

Modular Approach

The API has been designed in a highly modular fashion, resulting in code that is:

- Easy to understand
- Easy to maintain
- Easy to expand

Error Handling

Oracle Receivables APIs provide an extensive error-handling and error-reporting mechanism whereby all errors encountered in the Defaulting and Validation phases are reported and put on the message stack. The calling program can look up all error messages, or the first error message on the stack.

If only one error exists on the message stack, then you do not need to fetch the message from the stack because the message will return as one of the output parameters of the API routine.

Robust Validation

The validations that Oracle Receivables APIs perform are robust in nature. The APIs collect all encountered validation errors and put them on the message stack. The relevant entity handler is called only if no errors are reported during the Defaulting and Validation phases.

Debug Messages

Extensive debug messages have been incorporated to simplify the troubleshooting process when problems are encountered with any API.

Debug messages can be written to the log file by calling the appropriate routines described in Exception Handling and Result Messages, page 1-3.

Solution Outline

Modular Approach

To modularize an API, the basic structure of the API is divided into four parts:

1. Defaulting the IDs from the values and cross validating, if you provide both the values and the IDs.
2. Defaulting all the entity level information, which you have not entered or which the API needs internally.
3. Validating the entity level information that you entered.
4. Calling to the entity handlers to perform the relevant task.

This results in code that is easy to understand and easy to maintain. Any new functionality can be added by a simple code plug-in at each of the four parts.

Defaulting

In general, the various parameters in each API call get defaulted, if not entered, based on the following:

- Values of the other parameters in the API call
- Values set in the AR_SYSTEM_PARAMETERS table entered through the System Options form
- Relevant profile option values

Depending on the above three factors and the exact business requirement, the minimum number of parameters required to perform certain business tasks may vary.

Null values are defaulted for the parameters that could not be defaulted by the API defaulting routines.

For various attributes of the business objects, you can pass either the ID or the value of the attribute.

If you specify only the value, then the value is used to derive the ID; otherwise, the ID (if specified) is taken directly. If you specify both the ID and the value, then the ID takes precedence over the value and a warning message informs you of this.

Exception Handling and Result Messages

Each Oracle Receivables API returns three types of information to its calling programs:

- Overall status
- Messages describing the operations performed or errors encountered by the APIs
- Some output values that the API caller might want to use (this is different for different API routines and is described in each API's relevant chapter, in the API Usage section).

Return Status

The return status (`x_return_status`) of the API informs the caller about the result of the operation (or operations) performed by the API. The different possible values for an API return status are:

- Success (`FND_API.G_RET_STS_SUCCESS`)
- Error (`FND_API.G_RET_STS_ERROR`)
- Unexpected error (`FND_API.G_RET_STS_UNEXP_ERROR`)

The following section describes the different values of return status and their meanings.

Success

A success return status means that the API was able to perform all the operations requested by its caller. A success return status may be accompanied by informative messages in the API message list.

Error

An error return status means that the API failed to perform some or all of the operations requested by its caller. An error return status is usually accompanied by messages describing the error (or errors) and how to fix it.

In most cases, you should be able to take corrective action to fix regular, expected errors such as missing attributes or invalid date ranges.

Unexpected error

An unexpected error status means that the API has encountered an error condition it did not expect or could not handle. In this case, the API is unable to continue with its regular processing. Examples of such errors are irrecoverable data inconsistency errors, memory errors, and programming errors (such as attempting a division by zero).

In most cases, only system administrators or application developers can fix these unexpected errors.

Messages

The APIs put result messages into a message list. Programs calling the APIs can then get the messages from the list and process them by issuing them, loading them into a database table, or writing them to a log file.

Messages are stored in an encoded format to let the API callers find message names using the standard functions provided by the message dictionary. It also allows the storing of these messages in database tables and reporting off these tables in different languages.

The API message list must be initialized every time a program calls an API. API callers can either call the message list utility function `FND_MSG_PUB.Initialize` or request that the API do the initialization on their behalf by setting the `p_init_msg_list` parameter to

TRUE.

The program calling the API can retrieve messages from the message stack using the existing FND API functions FND_MSG_PUB.Count_Msg and FND_MSG_PUB.Get.

Message Level Threshold

The message level threshold is stored in a profile option named FND_API_MSG_LEVEL_THRESHOLD. This profile option can be updated at all levels (site, application, or user). The API checks against this threshold before writing a message to the API message list.

Debug Messages

You must enable debugging by calling the routine arp_standard.enable_file_debug. The routine requires 2 parameters: path_name and file_name.

```
arp_standard.enable_file_debug (<pathname>, <filename>)
```

The path name can be identified by using the following select statement:

```
select value from v$parameter where name = 'utl_file_dir',
```

The file name can be any name that you choose.

Example

```
arp_standard.enable_file_debug ('/sqlcom/log', 'txt.log')
```

This call would write the output debug file 'txt.log' in the path '/sqlcom/log'.

Calling Program Context

The program calling these APIs should have set up the application, responsibility, and user in the context of Oracle Application.

If the calling program does not set up this context, then it can be done programmatically by calling the following FND API.

```
fnd_global.apps_initialize ( user_id in number,  
                            resp_id in number,  
                            resp_appl_id in number,  
                            security_group_id in number default 0);
```

Adjustment API User Notes

Overview

This document outlines the use of the Adjustment API. This API allows users to create, approve, update, and reverse adjustments for invoices using simple calls to PL/SQL functions.

The Adjustment API is not intended to replace the existing Adjustment form, Adjustment Approval form, or the batch Auto-Adjust program.

Note: The Adjustment API requires the following receivable activity setup: the GL Account Source must be *Activity*.

You can access the API in two ways:

- With standard PL/SQL servers-side routine calls
- Through Forms, using the capability of Forms6 to have a procedure as its underlying base table.

API Usage

To create, modify, approve, or reverse adjustments, use the following routines:

- [Ar_Adjust_pub.Create_Adjustment](#), page 2-4: Use this routine to create an adjustment for an invoice.
- [Ar_Adjust_pub.Create_Linelevel_Adjustment](#), page 2-25: Use this routine to create an adjustment for an invoice/debit memo at line level.
 - Line level adjustments result in distributions being created only

for the invoice line being adjusted. Such distributions are not prorated across all invoice lines but are applicable only to the specified invoice line using the remaining balances of the invoice line.

- This routine creates the adjustment at the line level, and maintains the line level balances of the transactions.
- This routine is restricted to do line level adjustment for adjustment type – LINE. The tax amount is prorated based on the receivable activity tax code source passed for the line adjustment.
- Tax, freight and charges adjustments can only be done at the header level of the invoice. Use the standard Create_Adjustment API to create tax, freight and charges adjustments applicable to the invoice header.
- You cannot modify the line level adjustment using modify routine in the API. However, you can use the Reverse_Adjustment routine to reverse an existing adjustment by passing the applicable adjustment identifier.
- Receivables does not support line level adjustment functionality for invoices with installments. You can make adjustments against an invoice with multiple installments only at the invoice header level.
- You cannot create line level adjustments against Invoices or debit memos that were created prior to Release 12 and have activity against them.

- Ar_Adjust_pub.Modify_Adjustment, page 2-18: Use this routine to modify an adjustment's status, comments, and reason code.

Note: If the existing status of the adjustment is A or R, then it cannot be modified.

- Ar_Adjust_pub.Approve_Adjustment, page 2-13: Use this routine to approve an adjustment.
- Ar_Adjust_pub.Reverse_Adjustment, page 2-23: Use this routine to reverse an adjustment at either header or line level.

The Adjustment API has a defaulting mechanism for input parameters. This lets you create, approve, update, and reverse adjustments while passing a minimal number of API parameters.

Note: You can pass an adjustment account and override the default account (derived from the adjustment's receivable activity), provided that:

- The GL account source on the adjustment's receivable activity is *Activity* and the tax code source is *None*.
- The AR: Override Adjustment Activity Account Option profile option is set to *Yes*.

The following table shows standard API parameters that are common to all routines in the Adjustment API:

Parameter	Type	Data-type	Required	Default Value	Description
p_api_version	IN	NUMBER	Yes		Used to compare version numbers of incoming calls to its current version number.
p_init_msg_list	IN	VARCHAR2		FND_API.G_FALSE	Allows API callers to request that the API does initialization of the message list on their behalf.
p_commit	IN	VARCHAR2		FND_API.G_FALSE	Used by API callers to ask the API to commit on their behalf.
p_validation_level	IN	NUMBER		FND_API.G_VALID_LEVEL_FULL	Not currently for use by the user. Allow this parameter to default.
p_return_status	OUT	VARCHAR2			Represents the API overall return status. For possible values, see Error Handling, page 1-2.
p_msg_count	OUT	NUMBER			Number of messages in the API message list

Parameter	Type	Data-type	Required	Default Value	Description
p_msg_data	OUT	VARCHAR2			This is the message in encoded format if p_msg_count=1

Ar_Adjust_pub.Create_Adjustment

Use this routine to create adjustments to invoices. The API returns the Out parameter p_new_adjust_id, which represents the newly-created adjustment ID. The following is a breakdown of parameters for this routine, divided according to parameter type:

Input Parameters

Standard API parameters: 4

Create Adjustment parameters: 6 required parameters (might vary depending on the adjustment type)

Output Parameters

Standard API parameters: 3

Create Adjustment parameters: 2

Since the Create Adjustment API allows users to pass the adjustment record type to the procedure, it is not recommended that users enter values for unnecessary fields. These fields could be populated for internal use only.

Parameter Descriptions

The following table lists parameters that pertain specifically to the Create Adjustment routine:

Parameter	Type	Data-type	Required	Default Value	Description
p_adj_rec.type	IN	VARCHAR2	Yes		The type of adjustment to be created. Possible Values: 'INVOICE', 'LINE', 'TAX', 'FREIGHT', 'CHARGES', 'FINCHRG'
p_adj_rec.payment_schedule_id	IN	NUMBER	Yes		Payment Schedule id of the transaction for which the transaction is to be created.

Parameter	Type	Data-type	Required	Default Value	Description
p_adj_rec.amount	IN	NUMBER	Yes/No		If the adjustment type is any other value than 'INVOICE' then this is a required field. The amount indicates the amount to be adjusted.
p_adj_rec.customer_trx_line_id	IN	NUMBER	Yes/No		If the adjustment type is 'LINE' then the customer_trx_line_id indicates the line to be adjusted. For all the other adjustment types the value is not required.
p_adj_rec.receivables_trx_id	IN	NUMBER	Yes		The id of the activity name (from ar_receivables_trx) should be passed.
p_adj_rec.code_combination_id	IN	NUMBER	No		The code combination id is not required. If the value is not passed, then the default is the code combination id specified in the receivables_trx_id record. If the value passed is not the same as the code_combination_id and the AR: Override Adjustment Activity Account Option profile option is set to No, then this would error out.
p_adj_rec.apply_date	IN	DATE	Yes		The apply date should be equal to or greater than the transaction date.
p_adj_rec.gl_date	IN	DATE	Yes		The gl date should be equal to or greater than the transaction gl date, and the date should be from the open/future period.
p_adj_rec.reason_code	IN	VARCHAR 2	No		The reason code should a valid reason code in ar_lookups with lookup_type = 'ADJUST_REASON'.
p_adj_rec.comments	IN	VARCHAR 2	No		The user can enter comments, up to 2000 bytes, for creating the adjustments which could be useful for the user, for future reference.

Parameter	Type	Data-type	Required	Default Value	Description
p_adj_rec.associated_cash_receipt_id	IN	NUMBER	No		The associated cash receipt id is the id of a valid cash receipt, and is to be associated with the adjustment.
p_adj_rec.ussgl_transaction_code	IN	VARCHAR 2	No		The USSGL transaction code should be a valid USSGL transaction code in gl_ussgl_transaction_codes.
p_adj_rec.created_by	IN	VARCHAR 2	Yes		Some value that indicates to the user that it was created through the Adjustment API. Eg. 'ADJ-API'
p_adj_rec.attribute_category, p_adj_rec.attribute1 - p_adj_rec.attribute15	IN	VARCHAR 2	No		This attribute_category and the attribute1 through attribute15 can be entered if the user want to enter the details of the descriptive flexfield for the adjustment.
p_adj_rec.adjustment_id	IN		No. Entered values will be overwritten.		
p_adj_rec.acctd_amount	IN		No. Entered values will be overwritten.		
p_adj_rec.gl_posited_date	IN		No. Entered values will be overwritten.		

Parameter	Type	Data-type	Required	Default Value	Description
p_adj_rec.set_of_books_id	IN		No. Entered values will be overwritten.		
p_adj_rec.adjustment_type	IN		No. Entered values will be overwritten.		
p_adj_rec.status	IN		No. Entered values will be overwritten.		
p_adj_rec.line_adjusted	IN		No. Entered values will be overwritten.		
p_adj_rec.freight_adjusted	IN		No. Entered values will be overwritten.		
p_adj_rec.tax_adjusted	IN		No. Entered values will be overwritten.		

Parameter	Type	Data-type	Required	Default Value	Description
p_adj_rec.receiveables_charges_adjusted	IN		No.		Entered values will be overwritten.
p_adj_rec.batch_id	IN		No.		Entered values will be overwritten.
p_adj_rec.customer_trx_id	IN		No.		Entered values will be overwritten.
p_adj_rec.subsequent_trx_id	IN		No.		Entered values will be overwritten.
p_adj_rec.chargeback_customer_tx_id	IN		No.		Entered values will be overwritten.
p_adj_rec.distribution_set_id	IN		No.		Entered values will be overwritten.

Parameter	Type	Data-type	Required	Default Value	Description
p_adj_rec.associated_application_id	IN		No.		Entered values will be overwritten.
p_adj_rec.automatically_generate_id	IN		No.		Entered values will be overwritten.
p_adj_rec.postable_line	IN		No.		Entered values will be overwritten.
p_adj_rec.approved_by	IN		No.		Entered values will be overwritten.
p_adj_rec.adjustment_number	IN		No.		Entered values will be overwritten.
p_adj_rec.sequence_value	IN		No.		Entered values will be overwritten.

Parameter	Type	Data-type	Required	Default Value	Description
p_adj_rec.doc_se quence_id	IN		No. Entered values will be overwritten.		
p_adj_rec.postin g_control_id	IN		No. Entered values will be overwritten.		
p_adj_rec.last_u pdated_by	IN		No. Entered values will be overwritten.		
p_adj_rec.last_u pdated_date	IN		No. Entered values will be overwritten.		
p_adj_rec.last_u pdated_login	IN		No. Entered values will be overwritten.		
p_adj_rec.create d_by	IN		No. Entered values will be overwritten.		

Parameter	Type	Data-type	Required	Default Value	Description
p_adj_rec.creatio n_date	IN		No. Entered values will be overwritten.		
p_adj_rec.progra m_application_i d	IN		No. Entered values will be overwritten.		
p_adj_rec.progra m_id	IN		No. Entered values will be overwritten.		
p_adj_rec.progra m_update_date	IN		No. Entered values will be overwritten.		
p_adj_rec.reques t_id	IN		No.		
p_chk_approval _limits	IN	VARCHAR 2	No.	FND_API.G_ TRUE	This value can be set to 'F' if the adjusted amount should not be validated against the users approval limit.
p_move_deferre d_tax	IN	VARCHAR 2	No.	Y	This parameter is only used for BR.
p_check_amount	IN	VARCHAR 2	No.	FND_API.G_ TRUE	This value should never be set to 'F'. It is used for some internal logic.

Parameter	Type	Data-type	Required	Default Value	Description
p_new_adjust_number	OUT	ar_adjustment.adjustment_number%type			If the adjustment is created successfully, then this parameter will contain the value of the new adjustment number.
p_new_adjust_id	OUT	ar_adjustment.adjustment_id%type			If the adjustment is created successfully, then this parameter will contain the value of the new adjustment id.
p_called_from	IN	VARCHAR2	No	NULL	This flag is only used for BR.

Note: If the user passes values for any parameter not reported in the table above, then those values will be ignored and will not show up in the record.

Default values for API parameters derive from the following:

- Values of the other parameters in the API call
- Values set in the ar_system_parameters table entered through the System Options form
- Relevant profile option values

Depending on the user's particular business needs, the minimum number of parameters required to create an adjustment may vary.

Validation of the parameters passed

All the parameters that are passed to the API are validated, and if any of the required fields are missing or invalid, then the API returns an error message. A list of possible error messages appears in *Messages*, page 2-28.

Example

The following is the simplest test case for creating an adjustment.

Objective:

To create an adjustment, passing the minimum number of parameters.

Entered parameters:

```

p_adj_rec.type = 'INVOICE',
p_adj_rec.payment_schedule_id = 22222,
p_adj_rec.receivables_trx = 15,
p_adj_rec.apply_date = to_date('12-FEB-00', 'DD-MON-RR'),
p_adj_rec.gl_date = to_date('12-FEB-00', 'DD-MON-RR'),
p_adj_rec.created_from = 'ADJ-API'

```

Call to the API:

```

AR_ADJUST_PUB.Create_Adjustment(
    p_api_name          => 'AR_ADJUST_PUB',
    p_api_version       => 1.0,
    p_msg_count         => msg_count ,
    p_msg_data          => msg_data,
    p_return_status     => return_status,
    p_adj_rec           => adj_rec,
    p_new_adjust_number => new_adj_num,
    p_new_adjust_id     => new_adj_id );

```

Result:

Creates an adjustment, passing two standard required parameters and six adjustment record related parameters.

Ar_Adjust_pub.Approve_Adjustment

Use this routine to approve an adjustment. The following is a breakdown of parameters for this routine, divided according to parameter type:

Input Parameters

Standard API parameters: 4

Approve Adjustment parameters: 1 required parameter

Output Parameters

Standard API parameters: 3

Parameter Descriptions

Although the Approve Adjustments API allows users to pass the adjustment record type to the procedure, all the values are overwritten by the values in the existing adjustment record except for the status and gl_date.

The following table shows parameters that pertain specifically to the Approve Adjustment routine.

Note: If required parameters are not passed in a call to this API, then

the call will fail. If values are not required, then the values for those fields will be copied from the existing values of the adjustment.

Parameter	Type	Data-type	Required	Default Value	Description
p_old_adjust_id	IN	NUMBER	Yes		The id of the adjustment that needs to be approved.
p_adj_rec.type	IN	VARCHAR2	No		
p_adj_rec.payment_schedule_id	IN	NUMBER	No		
p_adj_rec.amount	IN	NUMBER	No		
p_adj_rec.customer_trx_line_id	IN	NUMBER	No		
p_adj_rec.receivable_trx_id	IN	NUMBER	No		
p_adj_rec.code_combination_id	IN	NUMBER	No		
p_adj_rec.apply_date	IN	DATE	No		
p_adj_rec.gl_date	IN	DATE	No	GL date of adjustment	The GL date should be entered if it is going to be different from the one in the old adjustment.
p_adj_rec.reason_code	IN	VARCHAR2	No		
p_adj_rec.comments	IN	VARCHAR2	No		
p_adj_rec.associated_cash_receipt_id	IN	NUMBER	No		

Parameter	Type	Data-type	Required	Default Value	Description
p_adj_rec.ussgl_t ransaction_code	IN	VARCHAR2	No		
p_adj_rec.create d_from	IN	VARCHAR2	No		
p_adj_rec.attribu te_category, p_adj_rec.attribu te1 - p_adj_rec.attribu te15	IN	VARCHAR2	No		
p_adj_rec.adjust ment_id	IN		No		
p_adj_rec.acctd_ amount	IN		No		
p_adj_rec.gl_pos ted_date	IN		No		
p_adj_rec.set_of_ books_id	IN		No		
p_adj_rec.adjust ment_type	IN		No		
p_adj_rec.status	IN		No	'A' if the status is null.	Possible Value: 'A' which indicates Approval
p_adj_rec.line_a djusted	IN		No		
p_adj_rec.freight _adjusted	IN		No		
p_adj_rec.tax_ad justed	IN		No		

Parameter	Type	Data-type	Require d	Default Value	Description
p_adj_rec.receiva bles_chages_adj usted	IN		No		
p_adj_rec.batch_ id	IN		No		
p_adj_rec.custo mer_trx_id	IN		No		
p_adj_rec.subseq uent_trx_id	IN		No		
p_adj_rec.charge back_customer_t rx_id	IN		No		
p_adj_rec.distrib ution_set_id	IN		No		
p_adj_rec.associa ted_application_ id	IN		No		
p_adj_rec.autom atically_generate d	IN		No		
p_adj_rec.postab le	IN		No		
p_adj_rec.appro ved_by	IN		No		
p_adj_rec.adjust ment_nu,mber	IN		No		
p_adj_rec.doc_se quence_value	IN		No		

Parameter	Type	Data-type	Required	Default Value	Description
p_adj_rec.doc_sequence_id	IN		No		
p_adj_rec.posting_control_id	IN		No		
p_adj_rec.last_updated_by	IN		No		
p_adj_rec.last_updated_date	IN		No		
p_adj_rec.last_updated_login	IN		No		
p_adj_rec.created_by	IN		No		
p_adj_rec.creation_date	IN		No		
p_adj_rec.program_application_id	IN		No		
p_adj_rec.program_id	IN		No		
p_adj_rec.program_update_date	IN		No		
p_adj_rec.request_id	IN		No		
p_chk_approval_limits	IN	VARCHAR2	No	FND_API.G_TRUE	This value can be set to 'F' if the adjusted amount should not be validated against the users approval limit.
p_move_deferred_tax	IN	VARCHAR2	No	Y	This flag is used only for Bills Receivable.

Validation of the parameters passed

All the parameters that are passed to the API are validated, and if any required fields are missing or invalid, then the API returns an error message. A list of possible error messages appears in *Messages*, page 2-28.

Example

The following is the simplest test case for approving an adjustment.

Objective:

To approve an adjustment, passing the minimum number of parameters.

Entered parameters:

adjustment_id = 88888;

Call to the API:

```
AR_ADJUST_PUB.Approve_Adjustment(
    p_api_name          => 'AR_ADJUST_PUB',
    p_api_version       => 1.0,
    p_msg_count         => msg_count ,
    p_msg_data          => msg_data,
    p_return_status     => return_status,
    p_old_adjust_id     => adjustment_id );
```

Result:

Approves an adjustment, passing 2 standard required parameters and 1 adjustment record parameter.

Ar_Adjust_pub.Modify_Adjustment

Use this routine to update an adjustment. The attributes that can be modified are comments, gl date, and status. If the status of the adjustment is already 'A' (i.e. the adjustment has already been approved), then you cannot update the adjustment. The following is a breakdown of parameters for this routine, divided according to parameter type:

Input Parameters

Standard API parameters: 4

Modify Adjustment parameters: 1 required parameter

Output Parameters

Standard API parameters: 3

Parameter Descriptions

Although the Modify Adjustments API allows users to pass the adjustment record type

to the procedure, all the values are overwritten by the existing adjustment record except for the status, comments, and gl_date.

The following table shows parameters that pertain specifically to the Modify Adjustments routine.

Note: If required parameters are not passed in a call to this API, then the call will fail. If values are not required, then the values for those fields will be copied from the existing values of the adjustment.

Parameter	Type	Data-type	Required	Default Value	Description
p_old_adjust_id	IN	NUMBER	Yes		The id of the adjustment that needs to be modified.
P_adj_rec.type	IN	VARCHAR2	No		
p_adj_rec.payment_schedule_id	IN	NUMBER	No		
p_adj_rec.amount	IN	NUMBER	No		
p_adj_rec.customer_trx_line_id	IN	NUMBER	No		
p_adj_rec.receivables_trx_id	IN	NUMBER	No		
p_adj_rec.code_combination_id	IN	NUMBER	No		
p_adj_rec.apply_date	IN	DATE	No		
p_adj_rec.gl_date	IN	DATE	No	GL date of adjustment	The GL date should be entered if the user wishes to modify the existing gl date of the adjustment.
P_adj_rec.reason_code	IN	VARCHAR2	No		

Parameter	Type	Data-type	Required	Default Value	Description
p_adj_rec.comment_s	IN	VARCHAR2	No		The comments should be entered if the user wishes to modify the existing comments of the adjustment.
P_adj_rec.associate_d_cash_receipt_id	IN	NUMBER	No		
p_adj_rec.ussgl_transaction_code	IN	VARCHAR2	No		
p_adj_rec.created_from	IN	VARCHAR2	No		
p_adj_rec.attribute_category,	IN	VARCHAR2	No		
p_adj_rec.attribute1					
p_adj_rec.attribute15					
p_adj_rec.adjustment_id	IN		No		
p_adj_rec.acctd_amount	IN		No		
p_adj_rec.gl_posted_date	IN		No		
p_adj_rec.set_of_books_id	IN		No		
p_adj_rec.adjustment_type	IN		No		
p_adj_rec.status	IN		No		The status should be entered if the user wishes to change the existing status of the adjustment. Possible Value: 'A', 'R', 'M', 'W'.
p_adj_rec.line_adjusted	IN		No		

Parameter	Type	Data-type	Required	Default Value	Description
p_adj_rec.freight_a djusted	IN		No		
p_adj_rec.tax_adju sted	IN		No		
p_adj_rec.receivabl es_chages_adjusted	IN		No		
p_adj_rec.batch_id	IN		No		
p_adj_rec.customer _trx_id	IN		No		
p_adj_rec.subseque nt_trx_id	IN		No		
p_adj_rec.chargeba ck_customer_trx_id	IN		No		
p_adj_rec.distributi on_set_id	IN		No		
p_adj_rec.associate d_application_id	IN		No		
p_adj_rec.automatic ally_generated	IN		No		
p_adj_rec.postable	IN		No		
p_adj_rec.approved _by	IN		No		
p_adj_rec.adjustme nt_nu, mber	IN		No		
p_adj_rec.doc_sequ ence_value	IN		No		

Parameter	Type	Data-type	Required	Default Value	Description
p_adj_rec.doc_sequence_id	IN		No		
p_adj_rec.posting_control_id	IN		No		
p_adj_rec.last_updated_by	IN		No		
p_adj_rec.last_updated_date	IN		No		
p_adj_rec.last_login	IN		No		
p_adj_rec.created_by	IN		No		
p_adj_rec.creation_date	IN		No		
p_adj_rec.program_application_id	IN		No		
p_adj_rec.program_id	IN		No		
p_adj_rec.program_update_date	IN		No		
p_adj_rec.request_id	IN		No		
p_chk_approval_limits	IN	VARCHAR2	No	FND_API.G_TRUE	This value can be set to 'F' if the adjusted amount should not be validated against the users approval limit.
p_move_deferred_tax	IN	VARCHAR2	No	Y	This flag is only used for Y.

Validations of the parameters passed

All the parameters that are passed to the API are validated, and if any of the required fields are missing or invalid, then the API returns an error message. A list of possible error messages appears in [Messages, page 2-28](#).

Example

The following is the simplest test case for updating an adjustment.

Objective:

To update an adjustment, passing the minimum number of parameters. For this example, assume the user wants to update comments.

Entered parameters:

old_adjustment_id = 88888

adj_rec.comments = 'This is the new comment'

Call to the API:

```
AR_Adjust_Pub.Create_Adjustment(  
    p_api_name          => 'AR_Adjust_Pub',  
    p_api_version       => 1.0,  
    p_msg_count         => msg_count ,  
    p_msg_data          => msg_data ,  
    p_return_status     => return_status,  
    p_adj_rec           => adj_rec ,  
    p_old_adjust_id     => old_adjustment_id );
```

Result:

Updates an adjustment, passing two standard required parameters and one adjustment record parameter. Users should also pass values for other parameters that the user wishes to update in the adjustment record.

Ar_Adjust_pub.Reverse_Adjustment

Use this routine to reverse an adjustment. The following is a breakdown of parameters for this routine, divided according to parameter type:

Input Parameters

Standard API parameters: 4

Reverse Adjustment parameters: 1 required parameter

Output Parameters

Standard API parameters: 3

Reverse Adjustment parameters: 1

Parameter Descriptions

The following table shows parameters that pertain specifically to the Reverse Adjustment routine:

Parameter	Type	Data-type	Required	Default Value	Description
p_old_adjust_id	IN	NUMBER	Yes		The id of the adjustment that needs to be modified.
p_comments	IN	VARCHAR 2	No		The user can specify any comments that should appear in the reverse adjustment.
p_reversal_gl_date	IN	DATE	No	Old adjustments gl date	The user can enter a gl date if he wishes it to be different from the old adjustments gl date.
p_reversal_date	IN	DATE	No	Old adjustments date	The user can enter a date if he wishes it to be different from the old adjustments date.
p_new_adj_id	OUT	NUMBER			
p_chk_approval_limits	IN	VARCHAR 2	No	FND_API.G_TRUE	This value can be set to 'F' if the adjusted amount should not be validated against the users approval limit.
p_move_deferred_tax	IN	VARCHAR 2	No	Y	This flag is used only for Bills Receivable.
p_called_from	IN	VARCHAR 2	No	NULL	This flag is used only for Bills Receivable.

Validation of the parameters passed

All the parameters that are passed to the API are validated, and if any of the required fields are missing or invalid, then the API returns an error message. A list of possible error messages appears in *Messages*, page 2-28.

Example

The following is the simplest test case for reversing an adjustment.

Objective:

To reverse an adjustment, passing the minimum number of parameters.

Entered parameters:

old_adjustment_id = 88888

Call to the API:

```
AR_ADJUST_PUB.Reverse_Adjustment(
  p_api_name      => 'AR_ADJUST_PUB',
  p_api_version   => 1.0,
  p_msg_count     => msg_count,
  p_msg_data      => msg_data,
  p_return_status  => return_status,
  p_old_adjust_id  => old_adjustment_id
  p_new_adj_id     => new_adjustment_id);
```

Result:

Reverses an adjustment, passing two standard required parameters and one adjustment record parameter.

Ar_Adjust_pub.Create_Linelevel_Adjustment

Use this routine to create adjustments to invoices at line level. This routine uses some parameters that are specific to this routine, and all the Standard API and other input and output parameters used by the Create_Adjustment routine described in Ar_Adjust_pub.Create_Adjustment, page 2-4.

Parameter Descriptions

The following table lists parameters that pertain specifically to the Create_Linelevel_Adjustment routine:

Parameter	Type	Data-type	Required	Default Value	Description
p_llca_adj_trx_li nes_tbl.customer _trx_line_id	IN	NUMBER	Yes		Customer_trx_line_id indicates the line to be adjusted.
p_llca_adj_trx_li nes_tbl.receivabl e_trx_id	IN	NUMBER	Yes		The ID of the activity name (from ar_receivables_trx) should be passed.
p_llca_adj_trx_li nes_tbl.line_amo unt	IN	NUMBER	Yes		The amount indicates the amount to be adjusted (Including Tax amount).

Parameter	Type	Data-type	Required	Default Value	Description
p_llca_adj_create _tbl_type.adjust me_number	OUT	NUMBER			If the adjustment is created successfully, then this parameter contains the value of the new adjustment number.
p_llca_adj_create _tbl_type.adjust ment_id	OUT	NUMBER			If the adjustment is created successfully, then this parameter contains the value of the new adjustment_id.
p_llca_adj_create _tbl_type.custom er_trx_line_id	OUT	NUMBER			If the adjustment is created successfully, then this parameter contains the value of the corresponding customer trx_line_id.

Note: You should populate the line level details on p_llca_adj_trx_lines_tbl for particular transactions, so that the system creates an adjustment for each line on p_llca_adj_trx_lines_tbl. You can retrieve the created adjustment details by using p_llca_adj_create_tbl_type as an out parameter.

The adjustment amount, receivable_trx_id, customer_trx_line_id are not required at the Header Level. The routine ignores these parameters and gives precedence to the p_llca_adj_trx_lines_tbl input parameters.

Validation of the parameters passed

All the parameters that are passed to the API are validated, and if any of the required fields are missing or invalid, then the API returns an error message. A list of possible error messages appears in *Messages*, page 2-28.

The API populates all error messages in the global temporary table ar_llca_adj_trx_errors_gt. Programs calling the API can then get the messages from this table and process them by issuing them, loading them into a database table, or writing them to a log file.

You need to check if a record exist in error table. If no records exist for a customer_trx_id and customer_trx_line_id, only then adjustment are created for a line against the transaction.

Example

The following is the simplest test case for creating an adjustment at the line level.

Objective:

To create an adjustment at line level, passing the minimum number of parameters.

Entered parameters:

```
p_adj_rec.type = 'LINE',
p_adj_rec.payment_schedule_id = 220175,
p_adj_rec.receivables_trx = 684090,
p_adj_rec.apply_date = to_date('10-FEB-2008', 'DD-MON-YYYY'),
p_adj_rec.gl_date = to_date('10-FEB-2008', 'DD-MON-YYYY'),
p_adj_rec.created_from = 'ADJ-API'
p_llca_adj_trx_lines_tbl(1).customer_trx_line_id := 1228207;
p_llca_adj_trx_lines_tbl(1).line_amount := -10;
p_llca_adj_trx_lines_tbl(1).receivables_trx_id := 1280;
p_llca_adj_trx_lines_tbl(2).customer_trx_line_id := 1228208;
p_llca_adj_trx_lines_tbl(2).line_amount := -20;
p_llca_adj_trx_lines_tbl(2).receivables_trx_id := 3095;
```

Call to the API:

```
AR_ADJUST_PUB.Create_Linelevel_Adjustment(
  p_api_name          => 'AR_ADJUST_PUB',
  p_api_version       => 1.0,
  p_msg_count         => msg_count,
  p_msg_data          => msg_data,
  p_return_status     => return_status,
  p_adj_rec           => adj_rec,
  p_llca_adj_trx_lines_tbl => p_llca_adj_trx_lines_tbl,
  p_move_deferred_tax => 'Y',
  p_llca_adj_create_tbl_type => p_llca_adj_create_tbl_type,
  p_called_from        => p_adj_rec.created_from,
  p_old_adjust_id      => NULL);
```

Note: The p_move_deferred_tax => 'Y' flag is used only for bills receivables.

Result:

Creates an adjustment for each line; and the output parameter p_llca_adj_create_tbl_type contains the adjustment information as described below:

```
p_llca_adj_create_tbl_type.customer_trx_line_id(1) = 1228207
p_llca_adj_create_tbl_type.adjustment_number(1) = 6032
p_llca_adj_create_tbl_type.adjustment_id(1) = 22012
p_llca_adj_create_tbl_type.customer_trx_line_id(2) = 1228208
```

`p_llca_adj_create_tbl_type.adjustment_number(2) = 6033`
`p_llca_adj_create_tbl_type.adjustment_id(2) = 22013`

Messages

The following table describes the possible messages returned by the Adjustment API.

Message Number	Message Name	Message Description	Additional Comments
42963	AR_AAPI_ADJ_AMOUNT_ZERO	No Adjustment amount passed.	
42964	AR_AAPI_ADR_ZERO_INVOICE	Cannot adjust, because the amount due in the Payment Schedule is zero, and the type specified is INVOICE.	
42965	AR_AAPI_APPLYDATE_LT_TRXDATE	The Apply date &APPLY_DATE is earlier than the transaction date &TRX_DATE.	
42966	AR_AAPI_DOC_SEQ_NOT_REQD	The specified document sequence: &DOCUMENT_SEQ is not required as the Unique Sequence Number profile option does not allow it.	
42967	AR_AAPI_GLDATE_INVALID_PERIOD	The GL date: &GL_DATE is not in an open or future enterable period.	
42968	AR_AAPI_GLDATE_LT_APPLYDATE	The GL date &GL_DATE is earlier than the apply date &APPLY_DATE.	
42969	AR_AAPI_GLDATE_LT_TRXGLDATE	The Adjustment GL date &GL_DATE is earlier than the transaction GL date &TRX_GL_DATE.	
42970	AR_AAPI_INVALID_ADJ_ID	Invalid adjustment ID: &ADJUSTMENT_ID specified.	
42971	AR_AAPI_INVALID_CODE_COMBINATION_ID	Invalid code combination ID: &CCID	

Message Number	Message Name	Message Description	Additional Comments
42972	AR_AAPI_INVALID_CREATE_STATUS	Invalid status: &STATUS passed during creation of Adjustment	
42973	AR_AAPI_INVALID_DESC_FLEX	Invalid Descriptive Flexfield has been provided.	
42974	AR_AAPI_INVALID_PAYSCHD	Invalid Payment Schedule ID: &PAYMENT_SCHEDULE_ID	
42975	AR_AAPI_INVALID_RECEIVABLES_TRX_ID	Invalid receivables trx ID: &RECEIVABLES_TRX_ID	
42976	AR_AAPI_INVALID_REASON_CODE	The reason code &REASON_CODE is invalid.	
42977	AR_AAPI_INVALID_RECEIPT_ID	Invalid Associated Cash Receipt ID &ASSOCIATED_CASH_RECEIPT_ID has been specified.	
42978	AR_AAPI_INVALID_RX_CLASS	Adjustment not allowed for transactions of class: &CLASS	
42979	AR_AAPI_INVALID_TYPE	Invalid type of adjustment: &TYPE	
42980	AR_AAPI_INVALID_USSGL_CODE	Invalid USSGL Transaction Code &USSGL_CODE has been specified	
42981	AR_AAPI_LINE_ID_FOR_NONLINE	Customer trx line ID: &CUSTOMER_TRX_LINE_ID passed for type = &TYPE	
42982	AR_AAPI_NO_APPLY_DATE	Apply date has not been specified	
42983	AR_AAPI_NO_APPROVAL_CODES	No valid approval codes exists for Adjustments in the Lookup table	
42984	AR_AAPI_NO_CCID	No valid code combinations exist for Adjustment	

Message Number	Message Name	Message Description	Additional Comments
42985	AR_AAPI_NO_CCID_F OR_ACTIVITY	No code combination id exists for receivables trx ID: &RECEIVABLES_TRX_ID and no code combination has been specified	
42986	AR_AAPI_NO_CHAN GE_OR_REVERSE	No changes allowed for Adjustment with &STATUS status	
42987	AR_AAPI_NO_CREAT ED_FROM	No values specified for the Created From attribute of the adjustment	
42988	AR_AAPI_NO_CUSTO MER_ID	No customer ID exists for payment schedule ID: &PAYMENT_SCHEDULE_ID	
42989	AR_AAPI_NO_CUSTO MER_TRX_ID	No customer trx id exists for payment schedule ID: &PAYMENT_SCHEDULE_ID	
42990	AR_AAPI_NO_CUSTO MER_TRX_LINEID	Invalid customer trx line id: &CUSTOMER_TRX_LINE_ID passed for customer trx id: &CUSTOMER_TRX_ID	
42991	AR_AAPI_NO_GL_DA TE	GL date has not been specified	
42992	AR_AAPI_NO_OPEN_ FUTURE_PERIOD	No valid open or future enterable GL periods exist for the ledger ID &SET_OF_BOOKS_ID	
42993	AR_AAPI_NO_REASO N_CODES	No valid reason codes exist for Adjustments in the Lookup table	
42994	AR_AAPI_NO_RECEIV ABLES_TRX	No valid receivables activity exists for Adjustments	
42995	AR_AAPI_NO_TYPE_C ODES	No valid type codes exists for Adjustments in the Lookup table	

Message Number	Message Name	Message Description	Additional Comments
42996	AR_AAPI_NO_USSGL_CODES	No valid USSGL Codes exist for Adjustment	
42997	AR_AAPI_OVERRIDE_CCID_DISALLOW	Override Activity profile option does not allow to override the Code Combination ID provided in the Receivables Activity	
42998	AR_AAPI_USSGL_CODE_DISALLOW	USSGL code is not allowed as the USSGL profile option does not allow it	
4667279	AR_LL_ADJ_INSTALL_NOT_ALLOWED	You cannot create a line-level adjustment for a transaction with installments	
4667280	AR_LL_ADJ_LEGACY_NOT_ALLOWED	You cannot create a line-level adjustment for a transaction with activity.	This is applicable to invoices and debit memos created in Release 11 <i>i</i> and having prior activity on them.
4667283	AR_ADJ_API_CUST_LINE_ID_IG	The line-level customer transaction line ID takes precedence over header-level customer transaction line ID.	If you passed the customer_trx_line_id in p_adj_rec and it is also passed for the line in the PLSQL table (used to pass line level adjustment data), this warning message is displayed.
4667282	AR_ADJ_API_AMOUNT_IG	The line-level amount takes precedence over the header-level amount.	If you passed the line_amount in p_adj_rec and it is also passed for the line in the PLSQL table (used to pass line level adjustment data), this warning message is displayed.
4667284	AR_ADJ_API_RECV_TRANSACTION_ID_IG	The line-level Receivables transaction ID takes precedence over the header-level Receivables transaction ID	If you passed the receivables_trx_id in p_adj_rec and it is also passed for the line in the PLSQL table (used to pass line level adjustment data), this warning message is displayed.
4667285	AR_ADJ_API_TYPE_DISALLOW	Receivables allows a line-level adjustment only for the line adjustment type	

Message Number	Message Name	Message Description	Additional Comments
4667281	AR_LL_ADJ MODIFY_ NOT_ALLOWED	You cannot modify an adjustment created at the line level	Use the reverse adjustment routine as stated in the earlier section of the documentation.

Credit Memo Approval and Creation API User Notes

Overview

This document outlines the use of the Credit Memo Approval and Creation API. This API lets you achieve the following task using simple calls to PL/SQL functions:

- Initiate a Credit Memo Request workflow process request for the creation of a credit memo against a specified transaction either with or without an approval process
- Check the status of an existing Credit Memo Request workflow process request
- Apply on-account credit memos to a debit item
- Unapply on-account credit memos to a debit item

To create a credit memo using an existing, user-defined Credit Memo Request workflow approval process, set the `p_skip_workflow_flag` parameter to `N`. In this case, the workflow process proceeds independently of the Credit Memo Approval and Creation API. If the disputed amount of the invoice is approved, then a credit memo is automatically created.

Note: You must set up the Credit Memo Request workflow before using the Credit Memo Approval and Creation API. For more information, see the *Oracle Receivables User Guide*.

To create a credit memo directly, without sending a request through the workflow approval process, set the `p_skip_workflow_flag` parameter to `Y`. If you set the `p_skip_workflow_flag` parameter to `Y`, then the Credit Memo Approval and Creation API bypasses the workflow process and calls code to automatically create the credit memo.

When you set the `p_skip_workflow_flag` parameter to `Y`, you might also have to set

values for its associated parameters: `p_credit_method_installments`, `p_credit_method_rules`, and `p_batch_source_name`. For more information, see the description of the `AR_CREDIT_MEMO_API_PUB.Create_Request` routine, page 3-2.

You cannot use the Credit Memo Approval and Creation API to generate on-account credit memos. You must specify an existing transaction to credit.

API Usage

To initiate a Credit Memo Request workflow process request, to check the status of an existing Credit Memo Request workflow process request, and to apply and unapply on-account credit memos to a debit item, use the following routines:

- `AR_CREDIT_MEMO_API_PUB.Create_Request`, page 3-2: Use this routine to initiate the Credit Memo Request workflow process by making a credit memo workflow request.
- `AR_CREDIT_MEMO_API_PUB.Get_Request_Status`, page 3-9: Use this routine to view the status of an existing request
- `AR_CM_API_PUB.Apply_On_Account`, page 3-13: Use this routine to apply on-account credit memos to a debit item.
- `AR_CM_API_PUB.Unapply_On_Account`, page 3-13: Use this routine to unapply on-account credit memo applied to a debit item.

Prerequisites

You must define three HTML pages that display this information:

- The credit memo dispute request
- The original transaction details
- The transaction activities

You provide the API with the URLs of these pages. When workflow notifications are sent to the collector, approver, and receivable roles, links to the URLs are set in the message body of the notification. If the URLs are not correctly set up, then you will receive an error message such as "URL not found" when you click on the links.

You must also set up the Credit Memo Request workflow before you use the Credit Memo Approval and Creation API. For more information, see: *Setting Up Credit Memo Request Workflow, Oracle Receivables User Guide*.

AR_CREDIT_MEMO_API_PUB.Create_Request

You can call this routine to create the Credit Memo Request workflow process request.

When the workflow request has been created, the API returns a unique request ID number (p_request_id) that you can use to track the status of the request. The following is a breakdown of this routine's parameters, based upon parameter type:

Standard Parameters

This table lists and describes the standard parameters common to all routines in the Credit Memo Approval and Creation API.

Parameter	Type	Data-type	Required	Default Value	Description
p_api_version	IN	NUMBER	Yes		Used to compare version numbers of incoming calls to current version number.
p_init_msg_list	IN	VARCHAR 2		FND_API.G_FA LSE	Set to TRUE to have the API automatically initialize the message list.
p_commit	IN	VARCHAR 2		FND_API.G_FA LSE	Set to TRUE to have the API commit automatically.
x_return_status	OUT	VARCHAR 2			Overall return status of the API.
x_msg_count	OUT	NUMBER			Number of messages in the API message list.
x_msg_data	OUT	VARCHAR 2			Message in encoded format if x_msg_count=1.

Create_Request Parameters

This table lists and describes parameters that specifically pertain to the Create_Request routine:

See Legend, page 3-8 for this table's legend.

Parameter	Type	Data-type	Required	Description
p_customer_trx_id	IN	ra_customer_trx.customer_trx_id %type	Yes	Customer_trx_id of the disputed invoice.

Parameter	Type	Data-type	Required	Description
p_line_credit_flag	IN	ra_cm_request.line_credit_flag%type	Yes	This value should be set to Y if the dispute is at the line level.
p_line_amount	IN	ra_cm_request.line_amount%type	Yes/No	Amount of the line dispute at the header level. If the dispute is at the header level, you should enter either the line_amount, tax_amount or freight_amount.
p_tax_amount	IN	ra_cm_request.tax_amount	Yes/No	Amount of the tax dispute at the header level.
p_freight_amount	IN	ra_cm_request.freight_amount	Yes/No	Amount of the freight dispute at header level.
p_cm_reason_code	IN	ra_cm_requests.cm_reason_code%type	YES	User defined lookup code that represents the reason for the invoice dispute. Should be a valid lookup_code for the lookup_type CREDIT_MEMO_REASON.
p_comments	IN	ra_cm_requests.comments%type	No	Comments about the credit memo request, entered if required. These comments appear in the notes region of the Transaction window. This also includes the internal comments posted by user and are not visible to customers.
p_orig_trx_number	IN	VARCHAR2	No	Enter the duplicate invoice number if using the "Duplicate Billing" reason code.
p_tax_ex_cert_num	IN	VARCHAR2	No	Tax exemption certificate number.

Parameter	Type	Data-type	Required	Description
p_request_url*	IN	VARCHAR2	No**	URL that displays the information of the actual credit memo dispute request.* See Legend, page 3-8 for this table's legend.
p_transaction_url	IN	VARCHAR2	No**	URL that displays the information of the original transaction. See Legend, page 3-8 for this table's legend.
p_trans_act_url	IN	VARCHAR2	No**	URL that displays information about the original transaction activities. See Legend, page 3-8 for this table's legend.
p_cm_line_tbl(x).customer_trx_line_id	IN	cm_line_tbl_type_cover%type	Yes/No	This value must be entered only if the dispute is at the line level. This value indicates the line_id that is in dispute. Note: Where p_cm_line_tbl(x), x indicates the index. The dispute can be for multiple lines.
p_skip_workflow_flag	IN	VARCHAR2	No	Defaults to N. If this value is set to Y, the entire workflow is skipped for that particular request and the credit memo is directly created.

Parameter	Type	Data-type	Required	Description
p_credit_method_ins	IN	VARCHAR2	No	<p>The p_credit_method_installments is the credit method that is used for crediting a transaction that uses split payment terms. Choices include PRORATE, LIFO, FIFO, or NULL.</p> <p>This value may be required if the p_skip_workflow_flag is set to Y.</p> <ul style="list-style-type: none"> • This parameter is mandatory if the credit memo is against a transaction that uses split payment terms and LINE_TYPE = LINE or CHARGES, or you are passing header freight. • Do not enter a value for this parameter if LINE_TYPE = TAX, or if you are passing freight for a specific line.

Parameter	Type	Data-type	Required	Description
p_credit_method_rul es	IN	VARCHAR2	No	<p>The p_credit_method_rules is the credit method for crediting a transaction which uses an accounting rule. Choices include PRORATE, LIFO, UNIT, or NULL.</p> <p>This value may be required if the p_skip_workflow_flag is set to Y.</p> <ul style="list-style-type: none"> • This parameter is mandatory if the credit memo is against a transaction which uses an accounting rule and LINE_TYPE = LINE or CHARGES, or you are passing header freight. • Do not enter a value for this parameter if LINE_TYPE = TAX, or if you are passing freight for a specific line.
p_batch_source_nam e	IN	VARCHAR2	No	This value is required if the p_skip_workflow_flag is set to Y.
p_org_id	IN	NUMBER	No	This value is required and it is used to identify the organization where the credit memo is sourced from.
x_request_id	OUT	VARCHAR2	Yes	Request_id of the credit memo that is returned if the data passed is valid and the credit memo request is created.
p_attribute_rec	IN	arw_cmreq_cover.pq_attribute_re c_type	No	Default value is ATTRIBUTE_REC_CONST.
p_interface_attribute _rec	IN	arw_cmreq_cover.pq_interface_re c_type	No	Default value is ATTRIBUTE_REC_CONST

Parameter	Type	Data-type	Required	Description
p_global_attribute_r ec	IN	arw_cmreq_cover.pq_global_attribute_rec_type	No	Default value is GLOBAL_ATTRIBUTE_REC_CONST.
P_DISPUTE_DATE	IN	DATE	No	NULL

Legend

* The request confirmation page might need the request_id as a parameter to query the information. This will not be available to the calling program when creating the p_request_url parameter because the request_id is the out parameter of the API. Calling programs should leave the request_id value blank and the table handler will add the request_id value and pass it to Workflow. The code searches for the "req_id=" string and replaces it with req_id="req_id". The parameter name must be req_id.

For example: For the old technology stack (PL/SQL), the following represents the request URL in *iReceivables* to call the "Request Confirmation" page. Note that no value has been entered for the req_id.

```
'arw_single_trx.single_cm_page?req_id='||'req_id='||'`&component='||glb_inv_part||'`&pct_change='||glb_percent_change;
```

** If the calling application does not enter the request, transaction, and transaction activities URLs, then you will see a default page reading "Unavailable" when you click on these links in the notifications screen. It is strongly recommended that the calling application have the UI (user interface) display these pages and pass these URLs to the API.

Parameter validation

The API validates all parameters that you enter. If any of the required fields are missing or invalid, then the API returns an error message. A list of error messages is documented in *Messages*, page 3-13.

Example

This example shows a simple test case for creating a credit memo request for a dispute at the header level:

Objective:

To create a credit memo request.

Parameters entered:

customer_trx_id = 99999

line_credit_flag = N

```

line_amount = -100
cm_reason_code = RETURN

```

Call to the API:

```

AR_CREDIT_MEMO_API_PUB.Create_Request(
    x_return_status      => p_return_status,
    x_msg_count         => p_msg_count,
    x_msg_data          => p_msg_data ,
CREDIT MEMO REQUEST PARAMETERS:
    p_customer_trx_id   => 99999,
    p_line_credit_flag  => 'N',
    p_line_amount        => -100,
    p_cm_reason_code    => 'RETURN',
    p_request_url       =>
    'arw_single_trx.single_trx_page?p1=19769&p2=1&wf=Y',
    p_transaction_url   =>
    'arw_single_trx.single_trx_page?p1=19769&p2=1&wf=Y'
    p_trans_act_url     =>
    'arw_single_trx.single_act_page?p1=19769&p2=1&wf=Y'
    x_request_id        => p_request_id

```

AR_CREDIT_MEMO_API_PUB.Get_Request_Status

Use this routine to view the Credit Memo Request workflow process request status. The API returns the status of the request and information about where the request is in the workflow. The following is a breakdown of parameters for this routine, based on parameter type:

Standard parameters

This table shows the standard API parameters common to all routines in the Credit Memo Approval and Creation API:

Parameter	Type	Data-type	Required	Default Value	Description
p_api_version	IN	NUMBER	Yes		Used to compare version numbers of incoming calls to its current version number.
p_init_msg_list	IN	VARCHAR2		FND_API_G_FALSE	Set to TRUE to have the API automatically initialize the message list.
x_return_status	OUT	VARCHAR2			Overall return status of the API.
x_msg_count	OUT	NUMBER			Number of messages in the API message list.

Parameter	Type	Data-type	Required	Default Value	Description
x_msg_data	OUT	VARCHAR2			Message, in encoded format if x_msg_count=1.

Get_Request_Status parameters

This table shows parameters that specifically pertain to the Get_Request_Status routine:

Parameter	Type	Data-type	Required	Description
p_request_id	IN	ra_cm_requests.request_id%type	YES	ID of the credit memo request whose status you are checking.
x_status_meaning	OUT	VARCHAR2		Status of the credit memo request.
x_reason_meaning	OUT	VARCHAR2		Reason for the dispute of the credit memo request.
x_customer_trx_id	OUT	ra_customer_trx.customer_trx_id%type		Customer transaction ID for the dispute of the credit memo request.
x_cm_customer_trx_id	OUT	ra_customer_trx.customer_trx_id%type		Credit memo transaction ID that was created for the dispute.
x_line_amount	OUT	ra_cm_requests.line_amount%type		Total amount of dispute for lines.
x_tax_amount	OUT	ra_cm_requests.tax_amount%type		Total amount of dispute for tax.
x_freight_amount	OUT	ra_cm_requests.freight_amount%type		Total amount of dispute for freight.
x_line_credits_flag	OUT	ra_cm_requests.line_credits_flag%type		Indicates whether the dispute is at the line level or the header level. If the value is set to Y, the dispute is at the line level.
x_created_by	OUT	wf_users.display_name%type		Name of the requestor.

Parameter	Type	Data-type	Required	Description
x_creation_date	OUT	DATE		Date of the request.
x_approval_date	OUT	DATE		Credit memo approval date if the credit memo has been created for the request.
x_comments	OUT	ra_cm_requests.comments% type		Comments entered by the requestor.
x_cm_line_tbl	OUT	cm_line_tbl_type_cover		Table that contains the line level dispute information. The values in the table will be set if the x_line_credits_flag = Y.
x_cm_activity_tbl	OUT	cm_activity_tbl_type_cover		Table that contains the status of the activities for the request.
x_cm_notes_tbl	OUT	cm_notes_tbl_type_cover		Table that contains the notes inserted for the transaction that is disputed.

Note:

```

TYPE CM_LINE_REC_TYPE_COVER IS RECORD
    customer_trx_line_id:
    ra_customer_trx_lines.customer_trx_line_id%type,
    extended_amount:  ra_customer_trx_lines.extended_amount%type,
    quantity_credited:  number,
    price:  number;
TYPE CM_LINE_TBL_TYPE_COVER
    IS TABLE OF
        CM_LINE_REC_TYPE_COVER
    INDEX BY BINARY INTEGER;
x_cm_line_tbl  CM_LINE_TBL_TYPE_COVER;
TYPE CM_ACTIVITY_REC_TYPE_COVER IS RECORD
    begin_date:  DATE,
    activity_name:  VARCHAR2(80),
    status:  wf_item_activity_statuses.activity_status%type,
    user:  wf_item_activity_statuses.activity_user%type);
TYPE CM_ACTIVITY_TBL_TYPE_COVER
    IS TABLE OF
        CM_ACTIVITY_REC_TYPE_COVER
    INDEX BY BINARY INTEGER;
x_cm_activity_tbl  CM_ACTIVITY_TBL_TYPE_COVER;
TYPE CM_NOTES_REC_TYPE_COVER IS RECORD
    ( NOTES  ar_notes.text%type);
TYPE CM_NOTES_TBL_TYPE_COVER
    IS TABLE OF
        CM_NOTES_REC_TYPE_COVER
    INDEX BY BINARY INTEGER;
x_cm_notes_tbl  CM_NOTES_TBL_TYPE_COVER;

```

Parameter validation

The API validates all parameters that you enter. If any of the required fields are missing or invalid, then the API returns an error message. A list of error messages is documented in [Messages](#), page 3-13.

Example

The following example is a simple test case for viewing the status of the credit memo request.

Objective:

To get the status of the credit memo request.

Parameters entered:

request_id = 122

Call to the API:

```

AR_CREDIT_MEMO_API_PUB.Get_Request_Status(
    p_api_version      => 1.0,
    x_msg_count        => msg_count,
    x_msg_data         => msg_data,
    x_return_status    => return_status,
    p_request_id       => request_id,
    x_status_meaning   => status_meaning,
    x_reason_meaning   => reason_meaning,
    x_customer_trx_id  => customer_trx_id,
    x_cm_customer_trx_id => cm_customer_trx_id,
    x_line_amount       => line_amount,
    x_tax_amount        => tax_amount,
    x_freight_amount   => freight_amount,
    x_line_credits_flag => line_credits_flag,
    x_created_by        => created_by,
    x_creation_date    => creation_date,
    x_cm_line_tbl       => cm_line_tbl,
    x_cm_activity_tbl   => cm_activity_tbl,
    x_cm_notes_tbl      => cm_notes_tbl);

```

AR_CM_API_PUB.Apply_On_Account

Use this routine to apply on-account credit memos to a debit item.

AR_CM_API_PUB.Unapply_On_Account

Use this routine to unapply on-account credit memo applied to a debit item.

Messages

The following table describes the possible messages returned by the Credit Memo Approval and Creation API.

Message Number	Message Name	Message Description
11936	AR_RAXTRX-1719	You must supply a reason code for your credit memo transaction.
11091	AR_CKAP_OVERAPP	You cannot overapply this transaction.
42711	AR_TAPI_LINE_NOT_EXIST	Line does not exist (customer_trx_line_id:[customer_trx_line_id]).
42756	AR_TAPI_TRANS_NOT_EXIST	Transaction does not exist (customer_trx_id:[customer_trx_id]).

Message Number	Message Name	Message Description
294003	AR_CMWF_API_INVALID_VAL UE	You specified an invalid value for the LINE_CREDIT_FLAG parameter. The valid values are Y and N.
294004	AR_CMWF_API_NO_LINES_INF O	The value for LINES_CREDIT_FLAG is Y, please provide at least one line level information.
294002	AR_CMWF_API_INVALID_REQ UEST_ID	Request does not exist (REQUEST_ID: &REQUEST_ID)

Credit Memo Application API User Notes

Overview

This document outlines in detail the Credit Memo Application API. This public API currently allows the application of an on-account credit memo to an activity such as Electronic Refund, which creates a customer refund for the credit memo via the creation of a negative miscellaneous receipt.

Please refer to the *Oracle Receivables User Guide* for more detail on customer credit refund functionality.

Basic Business Needs

The Credit Memo Application API enables the following business actions:

- Application of an on-account memo to the Electronic Refund activity, resulting in subsequent refund of the credit memo to the customer.

API Usage

To apply or unapply an on-account credit memo to an activity, call the following APIs:

- `ar_cm_application_pub.activity_application`, page 4-1: Applies an on-account credit memo to an activity.
- `ar_cm_application_pub.activity_unapplication`, page 4-11: Unapplies an on-account credit memo from an activity.

ar_cm_application_pub.activity_application

Use this routine to apply an on-account credit memo to an activity, such as Electronic Refund. The API returns the `receivable_application_id` of the `receivable_application` created.

Input Parameters

Standard API parameters: 4

Activity application parameters: 12 (including 1 descriptive flexfield parameter and 1 global descriptive flexfield parameter)

Output Parameters

Standard API parameters: 3

Activity application parameters: 4 required parameters (might vary depending on the adjustment type)

Parameter Descriptions

The input descriptive flexfield parameter is a record of type attribute_rec_type:

```
TYPE attribute_rec_type IS RECORD
  (p_attribute_category      IN VARCHAR2,
   p_attribute1              IN VARCHAR2,
   p_attribute2              IN VARCHAR2,
   p_attribute3              IN VARCHAR2,
   p_attribute4              IN VARCHAR2,
   p_attribute5              IN VARCHAR2,
   p_attribute6              IN VARCHAR2,
   p_attribute7              IN VARCHAR2,
   p_attribute8              IN VARCHAR2,
   p_attribute9              IN VARCHAR2,
   p_attribute10             IN VARCHAR2,
   p_attribute11             IN VARCHAR2,
   p_attribute12             IN VARCHAR2,
   p_attribute13             IN VARCHAR2,
   p_attribute14             IN VARCHAR2,
   p_attribute15             IN VARCHAR2);
```

The input global descriptive flexfield parameter is a record of type global_attribute_rec_type:

```

TYPE global_attribute_rec_type IS RECORD
  (p_global_attribute_category      IN VARCHAR2,
   p_global_attribute1            IN VARCHAR2,
   p_global_attribute2            IN VARCHAR2,
   p_global_attribute3            IN VARCHAR2,
   p_global_attribute4            IN VARCHAR2,
   p_global_attribute5            IN VARCHAR2,
   p_global_attribute6            IN VARCHAR2,
   p_global_attribute7            IN VARCHAR2,
   p_global_attribute8            IN VARCHAR2,
   p_global_attribute9            IN VARCHAR2,
   p_global_attribute10           IN VARCHAR2,
   p_global_attribute11           IN VARCHAR2,
   p_global_attribute12           IN VARCHAR2,
   p_global_attribute13           IN VARCHAR2,
   p_global_attribute14           IN VARCHAR2,
   p_global_attribute15           IN VARCHAR2,
   p_global_attribute16           IN VARCHAR2,
   p_global_attribute17           IN VARCHAR2,
   p_global_attribute18           IN VARCHAR2,
   p_global_attribute19           IN VARCHAR2,
   p_global_attribute20           IN VARCHAR2);

```

The following table lists the standard API parameters which are common to all the routines in the credit memo application API:

Parameter	Type	Data-type	Required	Default Value	Description
p_api_version	IN	NUMBER	Yes		Used to compare version numbers of incoming calls to its current version number.
					Unexpected error is raised if version in-compatibility exists.
					In the current version of the API, you should pass in a value of 1.0 for this parameter.
p_init_msg_list	IN	VARCHAR2		FND_API.G_FA_LSE	Allows API callers to request that the API does initialization of the message list on their behalf.
p_commit	IN	VARCHAR2		FND_API.G_FA_LSE	Used by API callers to ask the API to commit on their behalf.
p_validation_lev	IN	NUMBER	el	FND_API.G_V_ALID_LEVEL_FULL	Not to be used currently as this is a public API .

Parameter	Type	Data-type	Required	Default Value	Description
x_return_status	OUT	VARCHAR2			Represents the API overall return status. See: Exception Handling and Result Messages, page 1-3.
x_msg_count	OUT	NUMBER			Number of messages in the API message list
x_msg_data	OUT	VARCHAR2			This is the message in encoded format if x_msg_count=1

The following table lists the parameters that are relevant to the activity_application routine:

Parameter	Type	Data-type	Required	Description
p_customer_trx_id	IN	NUMBER	Yes	<p>Customer transaction identifier of the on account credit memo to be applied.</p> <p>Default: None</p> <p>Validation:</p> <p>Must be a completed on-account credit memo (not a regular credit memo related to an invoice where previous_customer_trx_id has a value).</p> <p>Must have a receipt method.</p> <p>Receipt method must belong to a receipt class that allows remittance.</p> <p>Receipt method must have a remittance bank account.</p> <p>Must have customer bank account details.</p> <p>Must have a negative original amount due.</p> <p>Must not be negative due to overpayment.</p> <p>Errors:</p> <p>AR_REF_NO_PAYMENT_METHODAR_REF_NO_CUST_BANK AR_REF_NOT_OACM AR_REF_CM_INCOMPLETE AR_REF_CM_POSITIVE AR_REF_MORE_THAN_CM_AMT AR_REF_RCT_CLASS_REMIT AR_REF_NO_REMIT_BANK</p>

Parameter	Type	Data-type	Required *	Description
p_amount_applied	IN	NUMBER	Yes	<p>The amount of the credit memo to apply to an activity.</p> <p>Default: None</p> <p>Validation:</p> <p>Must be greater than zero.</p> <p>Must not cause the credit memo to be overapplied.</p> <p>Total applied to Electronic Refund, including this application and other applications on the same credit memo, must not fall outside the approval limits or Credit Memo refunds specified for the user/currency.</p> <p>Errors:</p> <p>AR_CKAP_OVERAPP AR_REF_CM_APP_NEG AR_REF_USR_LMT_OUT_OF_RANGE</p>
p_applied_payment_schedule_id	IN	NUMBER	Yes	<p>The payment_schedule_id of the activity being applied to.</p> <p>Default: None</p> <p>Validation:</p> <p>Only 1 value is currently allowed: -8 (Electronic Refund)</p> <p>Error: AR_RAPI_APP_PS_ID_INVALID</p>
p_receivable_trx_id	IN	NUMBER	Yes	<p>Receivables Transaction Identifier of the activity being applied to.</p> <p>Default: None</p> <p>Validation: Must be of type Credit Memo Refund.</p> <p>Error: AR_RAPI_ACTIVITY_X_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_apply_date	IN	DATE	No	<p>The date the credit memo is applied to the activity.</p> <p>Default: The greater of system date and transaction date.</p> <p>Validation: Must not precede the transaction date of the credit memo.</p> <p>Error: AR_APPLY_BEFORE_TRANSACTION</p>
p_apply_gl_date	IN	DATE	No	<p>Date that this application will be posted to the General Ledger.</p> <p>Default: Gets defaulted to the system date if it is in an open or future enterable period, otherwise:</p> <ul style="list-style-type: none"> • If the most recent open period is prior to the transaction date: last date of that period • If there is a period open after the transaction date: first date of the last open period <p>Validation:</p> <p>It is valid if the following conditions are true:</p> <ul style="list-style-type: none"> • The date is in an Open or Future period • The period cannot be an Adjustment period • Must not precede the credit memo GL date <p>Error: AR_INVALID_APP_GL_DATE AR_VAL_GL_INV_GL</p>
p_ussgl_transaction_code	IN	VARCHAR2(30)	No	<p>Code defined by public sector accounting.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>

Parameter	Type	Data-type	Required *	Description
p_called_from	IN	VARCHAR2(20)	No	<p>This parameter is used to identify the calling routine.</p> <p>Default: Null</p> <p>Validation: None</p> <p>Error: None</p>
p_attribute_record	IN	attribute_rec_type	No	<p>This is a record type which contains all 15 descriptive flexfield segments and one descriptive flexfield structure defining column. It represents the Receivable Application Information flexfield.</p> <p>Default: DFF APIs used to do the defaulting</p> <p>Validation: DFF APIs used to do the validation depending on setup</p> <p>Error: AR_RAPI_DESC_FLEX_INVALID</p>
p_global_attribute_record	IN	global_attribute_rec_type	No	<p>This is a record type which contains all 20 global descriptive flexfield segments and one global descriptive flexfield structure defining column.</p> <p>Default: None</p> <p>Validation: None</p>
p_comments	IN	VARCHAR2(240)	No	User's comments
p_chk_approval_limit_flag	IN	VARCHAR2(1)	No	<p>Flag used to optionally override user approval limits for Credit Memo refunds.</p> <p>Values: 'Y' = Yes (check limits), 'N' = No (do not check limits)</p> <p>Default: 'Y'</p> <p>Validation: None</p>

Parameter	Type	Data-type	Required *	Description
p_application_ref_type	IN OUT	VARCHAR2(3 0)	No	<p>The context of application ref num/id is passed back in this parameter. For Electronic refunds this will be MISC_RECEIPT</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
P_application_ref_id	IN OUT	NUMBER	No	<p>For Electronic Refunds, the cash_receipt_id of the negative miscellaneous receipt created for the refund is passed back in this parameter.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_application_ref_nu m	IN OUT	VARCHAR2(3 0)	No	<p>For Electronic Refunds, the receipt number for the resulting miscellaneous receipt will be passed back in this parameter.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_receivable_applicati on_id	OUT	NUMBER(15)	No	The receivable application identifier of the activity application.

*If the values for the Required parameters are not passed in a call to this API, then the call itself will fail. However, depending on the business scenario, you must pass in values for other parameters to successfully create the business object. Otherwise, error messages will be reported.

Validation

This section explains the validation in this API, which does not relate directly to any of the above columns.

User Approval Limits

Only checked if p_chk_approval_limits_flag \neq 'N'.

- Limits must exist for the user calling the credit memo application API for the credit memo currency.
- The total amount applied to Electronic Refund for a particular credit memo must fall within the limits.

Example

Objective: To apply an on-account credit memo to the Electronic Refund activity using a call to ar_cm_application_pub.activity_application and passing the minimum number of input parameters.

This table lists the entered parameters:

Parameter	Entered Value	Default Value
p_api_version	1.0	
p_init_msg_list	FND_API.G_TRUE	
p_customer_trx_id	1001	
p_amount_applied	100	
p_applied_payment_schedule_id	1071	
p_receivables_trx_id	1089	

This table lists the defaulted input parameters, which were not entered:

Parameter	Entered Value	Default Value
p_apply_date		sysdate
p_apply_gl_date		sysdate
p_chk_approval_limit_flag		'Y'

The API call in this case would be:

```

Ar_cm_application_pub.activity_application(
    p_api_version          => 1.0,
    p_init_msg_list         => FND_API.G_TRUE,
    p_customer_trx_id      => 1001,
    p_amount_applied       => 100,
    p_applied_payment_schedule_id => 1071,
    p_receivables_trx_id   => 1089,
    x_return_status         => l_return_status,
    x_msg_count             => l_msg_count,
    x_msg_data              => l_msg_data,
    p_application_ref_type => l_application_ref_type,
    p_application_ref_id   => l_application_ref_id,
    p_application_ref_num  => l_application_ref_num);

```

The warnings and error messages put on the message stack by the API are retrieved after execution of this API by the calling program in the following manner:

```

IF l_msg_count = 1 Then
    --there is one message raised by the API, so it has been sent out
    --in the parameter x_msg_data, get it.
    l_msg_data_out := l_msg_data;
ELSIF l_msg_count > 1 Then
    --the messages on the stack are more than one so call them in a loop
    -- and put the messages in a PL/SQL table.
    loop
        count := count +1 ;
        l_mesg := FND_MSG_PUB.Get;
        If l_mesg IS NULL Then
            EXIT;
        else
            Mesg_tbl(count).message := l_mesg;
        End if;
    end loop;
END IF;

```

Depending on the message level threshold set by the profile option FND_API_MSG_LEVEL_THRESHOLD, the messages put on the message stack may contain both the error messages and the warnings.

Result: An electronic refund can be created for an on account credit memo by specifying only 6 input parameters in the above API call.

ar_cm_application_pub.activity_unapplication

Call this routine to reverse an activity application on an on-account credit memo. Such applications currently include only Electronic Refunds.

Input Parameters

Standard API parameters: 4

Activity unapplication parameters: 4

Output Parameters

Standard API parameters: 3

Activity unapplication parameters: 0

Parameter Descriptions

For descriptions of the seven standard API parameters, see `ar_cm_application_pub.activity_application`, page 4-1.

The following table lists the parameters that are relevant to the `activity_unapplication` routine:

Parameter	Type	Data-type	Required	Description
<code>p_customer_trx_id</code>	IN	NUMBER(15)	No	<p>The customer transaction identifier of the on-account credit memo from which the activity application is to be unapplied.</p> <p>Default: Null</p> <p>Validation:</p> <ol style="list-style-type: none">1. Must have at least one Electronic Refund application2. Must have only 1 Electronic Refund application if <code>receivable_application_id</code> is not supplied3. Must be specified if <code>receivable_application_id</code> is not specified <p>Error: AR_RAPI_CUST_TRX_ID_INVALID AR_RAPI_MULTIPLE_ACTIVITY_APP</p>

Parameter	Type	Data-type	Required	Description
p_receivable_application_id	IN	NUMBER(15)	No	<p>Identifies the receivable application. Used to derive the customer trx id if not specified.</p> <p>Default: If only one Electronic Refund application exists, then receivable-application_id is taken from it.</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. applied_payment_schedule_id must be -8 2. Display flag = 'Y' (latest application) and status = 'ACTIVITY' 3. Must correspond to the customer_trx_id specified. <p>Error: AR_RAPI_REC_APP_ID_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_reversal_gl_date	IN	DATE	*	<p>The reversal gl date used for the accounting entries.</p> <p>Default: Gets defaulted to the application gl date if it is a valid gl_date, otherwise:</p> <ul style="list-style-type: none"> • If the most recent open period is prior to the transaction date: last date of that period • If there is a period open after the transaction date: first date of the last open period <p>Validation:</p> <ol style="list-style-type: none"> 1. It is valid if the following conditions are true: <ul style="list-style-type: none"> • The date is in an Open or Future period • The period cannot be an Adjustment period 2. reversal GL date \geq application GL date 3. reversal GL date \geq transaction GL date <p>Error: AR_INVALID_APP_GL_DATE AR_RW_BEFORE_APP_GL_DATE AR_VAL_GL_INV_GL</p>
p_called_from	IN	VARCHAR2(20)	Yes	<p>Used to indicate which program is calling this API.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>

Example

Objective: To unapply an activity application using a call to ar_cm_application_pub.activity_unapplication, and passing a minimum number of input parameters.

This table lists the entered parameters:

Parameter	Entered value	Default Value
p_api_version	1.0	
p_receivable_application_id	10051	

This table lists the defaulted input parameters, which were not entered:

Parameter	Entered Value	Defaultvalue
p_customer_trx_id		20338
p_reversal_gl_date		01-JUN-2000
p_called_from		

The retrieval and handling of the warnings and error messages, put on the message stack by the API during execution, are the same as described in Exception Handling and Result Messages, page 1-3.

Messages

The following table lists all the error messages raised by the Credit Memo Application API:

TYPE

E: Error message

W: Warning message

MESSAGE_CODE	MESSAGE_TEXT	EXPLANATION	TYPE
AR_APPLY_BEFORE_TRANSACTION	Apply Date must be greater than or equal to the Transaction Date.		E
AR_CKAP_OVERAPP	You cannot over apply this transaction.	This message will appear if the amount being applied to the credit memo results in a change of sign of the balance due remaining	E

MESSAGE_CODE	MESSAGE_TEXT	EXPLANATION	TYPE
AR_INVALID_APP_GL_DATE	GL date, &GL_DATE, is not in an open or future-enterable period.	Either the GL date must be changed, or the period in which it falls must be opened or made future-enterable.	E
AR_RAPI_ACTIVITY_INVALID	The receivables activity name is invalid.		E
AR_RAPI_ACTIVITY_X_INVALID	The specified combination of payment schedule identifier and receivables transaction identifier is invalid.	The activity type derived from the receivables_trx_id does not match with the activity type of the specified payment_schedule_id.	E
AR_RAPI_APP_PS_ID_INVALID	Applied payment schedule identifier has an invalid value.		E
AR_RAPI_APP_PS_RA_ID_X_INVALID	Invalid receivable application identifier for the specified applied payment schedule identifier.		E
AR_RAPI_APPLIED_AMT_NULL	Applied amount could not be defaulted.	The p_applied_amount was not specified by the user and it could not be defaulted from the specified transaction	E
AR_RAPI_CUST_TRX_ID_INVALID	Invalid customer transaction identifier.		E
AR_RAPI_CUST_TRX_ID_NULL	Customer transaction identifier is null.		E
AR_RAPI_DESC_FLEX_INVALID	The entered values for the descriptive flexfield &DFF_NAME is invalid.		E
AR_RAPI_RCT_MD_ID_INVALID	Invalid receipt method identifier.		E
AR_RAPI_REC_APP_ID_INVALID	Invalid receivable application identifier.		E

MESSAGE_CODE	MESSAGE_TEXT	EXPLANATION	TYPE
AR_RAPI_REC_APP_ID_NULL	Receivable application identifier is null.		E
AR_RAPI_REC_TRX_ID_INVALID	Invalid receivable transaction identifier.		E
AR_RAPI_REC_TRX_ID_NULL	Please enter a receivables transaction identifier.		E
AR_RAPI_REV_GL_DATE_NULL	Reversal GL date is null.		E
AR_RAPI_TRX_PS_ID_X_INVALID	Invalid applied payment schedule identifier for the specified transaction.	The p_applied_payment_schedule_id specified by the user does not match with the payment_schedule_id derived from the p_customer_trx_id and the p_installment.	E
AR_RAPI_TRX_PS_NOT_DEF_CUS	The customer could not be defaulted from the entered transaction and the applied payment schedule identifier.		E
AR_RAPI_TRX_RA_ID_X_INVALID	The activity type for the entered receivable transaction identifier does not match with the activity of the entered payment schedule identifier .		E
AR_REF_BEFORE_CM_GL_DATE	The GL date cannot be before the credit memo GL date.		E
AR_REF_CM_APP_NEG	Only positive credit memo refund amounts are allowed.		E
AR_REF_CM_INCOMPLETE	Please complete this credit memo.		E
AR_REF_CM_POSITIVE	Credit memo refunds are only allowed on negative credit memos.		E

MESSAGE_CODE	MESSAGE_TEXT	EXPLANATION	TYPE
AR_REF_MORE_THAN_CM_AMT	You cannot refund more than the credit memo amount.		E
AR_REF_NO_CUST_BANK	To enable credit memo refunds, please add customer bank details to the credit memo.		E
AR_REF_NO_APPROVAL_LIMIT	Credit memo refund approval limits do not exist for this user and currency.		E
AR_REF_NO_PAYMENT_METHOD	To enable credit memo refunds, please add a receipt method to the credit memo.		E
AR_REF_NO_REMIT_BANK	To enable credit memo refunds, the credit memo receipt method must belong to a receipt class with an assigned remittance bank.		E
AR_REF_NOT_OACM	Credit memo refunds are only allowed for on-account credit memos.	Only transactions with Credit Memo class and unattached to another transaction are allowed.	E
AR_REF_RCT_CLASS_REMIT	To enable credit memo refunds, the receipt class associated with this credit memo receipt method must require remittance.		E
AR_REF_USR_LMT_OUT_OF_RANGE	The total refund amount must be within &FROM_AMOUNT and &TO_AMOUNT.		E
AR_RW_BEFORE_APP_GL_DATE	Reversal GL Date must be on or after original GL Date of &GL_DATE.		E
AR_VAL_GL_INV_GL	The GL date should not be prior to the invoice's GL date.		E

Deposit API User Notes

Overview

This document outlines the specifications and the methodology for using the various Commitment (Deposit) APIs. These APIs provide an extension to existing functionality of creating and manipulating deposits through the standard Oracle Receivables Transactions workbench.

You can access these APIs:

- As standard PL/SQL servers-side routine calls
- Through Forms, utilizing the capability of Forms6 to have a procedure as its underlying base table

Basic Business Needs

The Commitment (Deposit) API provides the following basic functionality via different API calls:

- Creates a commitment of type Deposit
- Creates non-revenue sales credit for a deposit

API Usage

To create a deposit, you can call the following PL/SQL APIs:

- AR_DEPOSIT_API_PUB.Create_deposit, page 5-2: Creates a single deposit and completes it.
- AR_DEPOSIT_API_PUB.insert_non_rev_salescredit, page 5-27: Creates nonrevenue sales credit for a deposit.

AR_DEPOSIT_API_PUB.Create_deposit

This routine is called to create a deposit for the transactions.

Only one owner can be assigned to a commitment.

This API routine has 8 output and 136 input parameters in total. Of the output parameters, the API returns CUSTOMER_TRX_ID, CUSTOMER_TRX_LINE_ID, and new TRX_NUMBER, if generated during deposit creation.

The following is the breakdown of the parameters:

Input

Standard API parameters: 4

Deposit parameters: 132 + 2 (global descriptive flexfield parameter)

Output

Standard API parameters: 3

Deposit parameters: 5

Parameter Descriptions

The input global descriptive flexfield parameter is a record of type global_attr_rec_type.

```

TYPE global_attr_rec_type IS RECORD(
    global_attribute_category      VARCHAR2(30) default null,
    global_attribute1              VARCHAR2(150) default NULL,
    global_attribute2              VARCHAR2(150) DEFAULT NULL,
    global_attribute3              VARCHAR2(150) DEFAULT NULL,
    global_attribute4              VARCHAR2(150) DEFAULT NULL,
    global_attribute5              VARCHAR2(150) DEFAULT NULL,
    global_attribute6              VARCHAR2(150) DEFAULT NULL,
    global_attribute7              VARCHAR2(150) DEFAULT NULL,
    global_attribute8              VARCHAR2(150) DEFAULT NULL,
    global_attribute9              VARCHAR2(150) DEFAULT NULL,
    global_attribute10             VARCHAR2(150) DEFAULT NULL,
    global_attribute11             VARCHAR2(150) DEFAULT NULL,
    global_attribute12             VARCHAR2(150) DEFAULT NULL,
    global_attribute13             VARCHAR2(150) DEFAULT NULL,
    global_attribute14             VARCHAR2(150) DEFAULT NULL,
    global_attribute15             VARCHAR2(150) DEFAULT NULL,
    global_attribute16             VARCHAR2(150) DEFAULT NULL,
    global_attribute17             VARCHAR2(150) DEFAULT NULL,
    global_attribute18             VARCHAR2(150) DEFAULT NULL,
    global_attribute19             VARCHAR2(150) DEFAULT NULL,
    global_attribute20             VARCHAR2(150) DEFAULT NULL,
    global_attribute21             VARCHAR2(150) DEFAULT NULL,
    global_attribute22             VARCHAR2(150) DEFAULT NULL,
    global_attribute23             VARCHAR2(150) DEFAULT NULL,
    global_attribute24             VARCHAR2(150) DEFAULT NULL,
    global_attribute25             VARCHAR2(150) DEFAULT NULL,
    global_attribute26             VARCHAR2(150) DEFAULT NULL,
    global_attribute27             VARCHAR2(150) DEFAULT NULL,
    global_attribute28             VARCHAR2(150) DEFAULT NULL,
    global_attribute29             VARCHAR2(150) DEFAULT NULL,
    global_attribute30             VARCHAR2(150) DEFAULT NULL);

```

The following table lists standard API parameters that are common to all the routines in the Commitment (Deposit) API.

Note: If required parameters are not passed in a call to this API, then the call will fail. However, depending on the business scenario, you will have to pass in values for other parameters to successfully create the business object; otherwise, error messages will be reported.

Parameter	Type	Data-type	Required	Default Value	Description
p_api_version	IN	NUMBER	Yes		Used to compare version numbers of incoming calls to its current version number. Unexpected error is raised if version incompatibility exists. In the current version of the API, you should pass a value of 1.0 for this parameter.

Parameter	Type	Data-type	Required	Default Value	Description
p_init_msg_list	IN	VARCHAR2		FND_API.G_FALSE	Allows API callers to request that the API does initialization of the message list on their behalf.
p_commit	IN	VARCHAR2		FND_API.G_FALSE	Used by API callers to ask the API to commit on their behalf.
p_validation_level	IN	NUMBER		FND_API.G_VALIDATION_LEVEL_FULL	Not to be used currently as this is a public API.
x_return_status	OUT	VARCHAR2			Represents the API overall return status.
x_msg_count	OUT	NUMBER			Number of messages in the API message list.
x_msg_data	OUT	VARCHAR2			This is the message in encoded format if x_msg_count=1.

The following table lists the parameters that pertain specifically to the deposit.

Parameter	Type	Data-type	Required*	Description
p_deposit_number	IN	VARCHAR2		<p>The deposit number of the deposit to be created.</p> <p>Default: Null</p> <p>Validation: If AR_RA_BATCH_AUTO_NUM_FLAG set by batch source is true, then it is derived automatically; else it is required to be present.</p> <p>Error: AR_DAPI_DEPOSIT_NO_NULL</p>
p_deposit_date	IN	DATE		<p>The deposit date of the entered deposit.</p> <p>Default: System date</p> <p>Validation: This field is mandatory.</p> <p>Error: None</p>

Parameter	Type	Data-type	Required*	Description
p_usr_currency_code	IN	VARCHAR2		<p>The translated currency code. Used to derive the p_currency_code if it is not entered.</p> <p>Default: None</p> <p>Validation: Should be a valid currency, so that the corresponding currency code can be derived.</p> <p>Error: AR_RAPI_USR_CURR_CODE_INVALID</p>
p_currency_code	IN	VARCHAR2		<p>The actual currency code that gets stored in AR tables.</p> <p>Default: Derived from p_usr_currency_code if entered, else defaults to the functional currency code.</p> <p>Validation: Validated against the currencies in FND_CURRENCIES table.</p> <p>Error: AR_RAPI_CURR_CODE_INVALID</p> <p>Warning: AR_RAPI_FUNC_CURR_DEFAULTED</p>
p_usr_exchange_rate_type	IN	VARCHAR2		<p>The translated exchange rate type. Used to derive the p_exchange_rate_type if it has not been entered.</p> <p>Default: None</p> <p>Validation: Should be a valid rate type.</p> <p>Error: AR_RAPI_USR_X_RATE_TYP_INVALID</p>

Parameter	Type	Data-type	Required*	Description
p_exchange_rate_ty pe	IN	VARCHAR2		<p>Exchange rate type stored in AR tables.</p> <p>Default: In case of foreign currency receipt, the value is derived from p_usr_exchange_rate_type. If p_usr_exchange_rate_type is null, then the value defaults from the AR: Default Exchange Rate Type profile option.</p> <p>Validation: Validated against values in GL_DAILY_CONVERSION_TYPES table.</p> <p>Error: AR_RAPI_X_RATE_TYPE_INVALID</p>
p_exchange_rate	IN	NUMBER		<p>The exchange rate between the receipt currency and the functional currency.</p> <p>Default: Derived from the Daily Rates table for rate_type <> User in case of nonfunctional currency. If Journals: Display Inverse Rate profile option = Y, set user-entered value to 1/ p_exchange_rate. The entered value is rounded to a precision of 38.</p> <p>Validation: In case of nonfunctional currency, the rate should have a positive value for rate type=User For nonfunctional currency and type is <> User, do not specify any value.</p> <p>Error: AR_RAPI_X_RATE_INVALID AR_RAPI_X_RATE_NULL</p>
p_exchange_rate_dat e	IN	DATE		<p>The date on which the exchange rate is valid.</p> <p>Default: Receipt date</p> <p>Validation: For a nonfunctional currency and type is <>User, there should be a valid rate existing in the database for this date. This is a cross validation of type, currency, and date.</p> <p>Error: AR_NO_RATE_DATA_FOUND</p>

Parameter	Type	Data-type	Required*	Description
p_batch_source_id	IN	NUMBER		<p>Batch source identifier for the commitment.</p> <p>Default: Same as ar_ra_batch_source profile option.</p> <p>Validation: It should be a valid batch source and it should exist in the database. This field is mandatory if not defined in profile option.</p> <p>Error: AR_DAPI_BS_NAME_INVALID AR_DAPI_BS_NAME_IGN AR_DAPI_BS_ID_INVALID</p>
p_batch_source_name	IN	VARCHAR2		<p>Batch source name for the commitment.</p> <p>Default: Same as ar_ra_batch_source_name profile option.</p> <p>Validation: It should be a valid batch source and it should exist in the database.</p> <p>Error: AR_DAPI_BS_NAME_INVALID AR_DAPI_BS_NAME_IGN AR_DAPI_BS_ID_INVALID</p>
p_cust_trx_type_id	IN	NUMBER		<p>Transaction Type identifier.</p> <p>Default: Based on the value of batch source</p> <p>Validation: It should be a valid transaction type. This field is mandatory.</p> <p>Error: AR_DAPI_TRANS_TYPE_INVALID AR_RAPI_TRANS_TYPE_IGN AR_DAPI_TRANS_TYPE_ID_INVALID</p>
p_cust_trx_type	IN	VARCHAR2		<p>Transaction Type name.</p> <p>Default: Based on the value of batch source</p> <p>Validation: It should be a valid transaction type.</p> <p>Error: AR_DAPI_TRANS_TYPE_INVALID AR_RAPI_TRANS_TYPE_IGN AR_DAPI_TRANS_TYPE_ID_INVALID</p>

Parameter	Type	Data-type	Required*	Description
p_class	IN	VARCHAR2		<p>Constant value = DEP. Keeping as an input for a future enhancement.</p>
p_gl_date	IN	DATE		<p>Date that this deposit will be posted to the general ledger.</p> <p>Default: Gets defaulted to the current date if it is a valid gl_date, otherwise:</p> <ul style="list-style-type: none"> • If the most recent open period is prior to the receipt date: last date of that period. • If there is a period open after the deposit date: first date of the last open period. <p>Validation: The gl date is valid if the following conditions are true:</p> <ul style="list-style-type: none"> • The date is in an Open or Future period. • The period cannot be an Adjustment period. <p>Error: AR_INVALID_APP_GL_DATE</p>

Parameter	Type	Data-type	Required*	Description
p_bill_to_customer_id	IN	NUMBER		<p>The CUSTOMER_ID for the bill-to customer.</p> <p>Default: Defaulted from customer name/number. If all name, number, and ID are null, then it is same as ship-to CUSTOMER_ID.</p> <p>Validation: Customer exists and has prospect code = CUSTOMER. Customer has a profile defined at the customer level. Either bill-to or ship-to customer must exist.</p> <p>Error: AR_RAPI_CUST_ID_INVALID AR_RAPI_CUS_NAME_INVALID AR_RAPI_CUS_NUM_INVALID AR_RAPI_CUS_NAME_NUM_INVALID AR_RAPI_CUS_NAME_NUM_IGN AR_DAPI_BILL_OR_SHIP_CUST_REQ</p>
p_bill_to_customer_name	IN	VARCHAR2		<p>The name for the entered customer. Used to default the customer ID if not specified.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPI_CUS_NAME_INVALID</p>
p_bill_to_customer_number	IN	VARCHAR2		<p>The number for the entered customer. Used to default the customer ID if not specified.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPI_CUS_NAME_INVALID</p>
p_bill_to_location	IN	VARCHAR2		<p>The location for the bill-to customer.</p> <p>Default: Defaulted from the primary bill-to customer location, if defined. Otherwise, null.</p> <p>Validation: This field is mandatory.</p> <p>Error: AR_DAPI_CUS_LOC_INVALID</p>

Parameter	Type	Data-type	Required*	Description
p_bill_to_contact_id	IN	NUMBER		<p>The contact identifier for the bill-to customer.</p> <p>Default: Defaulted from the bill-to customer site level, then customer level, if defined. Otherwise, null.</p> <p>Validation: Yes</p> <p>Error: AR_DAPI_BILL_CONTACT_NAME_INV AR_DAPI_CUS_CONTACT_INVALID</p>
p_bill_to_contact_fir st_name	IN	VARCHAR2		<p>The first name of contact for the bill-to customer.</p> <p>Default: Defaulted from bill-to customer site level, then customer level, if defined. Otherwise, null.</p> <p>Validation: This field is mandatory.</p> <p>Error: AR_DAPI_BILL_CONTACT_NAME_INV AR_DAPI_CUS_CONTACT_INVALID</p>
p_bill_to_contact_las t_name	IN	VARCHAR2		<p>The last name of contact for the bill-to customer.</p> <p>Default: Defaulted from bill-to customer site level, then customer level, if defined. Otherwise, null.</p> <p>Validation: This field is mandatory.</p> <p>Error: AR_DAPI_BILL_CONTACT_NAME_INV AR_DAPI_CUS_CONTACT_INVALID</p>

Parameter	Type	Data-type	Required*	Description
p_ship_to_customer_id	IN	NUMBER		<p>The CUSTOMER_ID for the ship-to customer.</p> <p>Default: Defaulted from customer name/number. Null otherwise.</p> <p>Validation: Customer exists and has prospect code = CUSTOMER. Customer has a profile defined at the customer level. Either bill-to or ship-to customer must exist.</p> <p>Error: AR_RAPI_CUST_ID_INVALID AR_RAPI_CUS_NAME_INVALID AR_RAPI_CUS_NUM_INVALID AR_RAPI_CUS_NAME_NUM_INVALID AR_RAPI_CUS_NAME_NUM_IGN AR_DAPI_BILL_OR_SHIP_CUST_REQ</p>
p_ship_to_customer_name	IN	VARCHAR2		<p>The name for the entered customer. Used to default the customer ID, if not specified.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPI_CUS_NAME_INVALID</p>
p_ship_to_customer_number	IN	VARCHAR2		<p>The number for the entered customer. Used to default the customer ID, if not specified.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPI_CUS_NAME_INVALID</p>
p_ship_to_location	IN	VARCHAR2		<p>The location for the bill-to customer.</p> <p>Default: Defaulted from primary bill-to customer location, if defined. Otherwise, null.</p> <p>Validation: This field is mandatory.</p> <p>Error: AR_DAPI_CUS_LOC_INVALID</p>

Parameter	Type	Data-type	Required*	Description
p_ship_to_contact_id	IN	NUMBER		<p>The contact identifier for the bill-to customer.</p> <p>Default: Defaulted from bill-to customer site level, then from customer level, if it is defined. If not defined, then it is not defaulted.</p> <p>Validation: Yes</p> <p>Error: AR_DAPI_BILL_CONTACT_NAME_INV AR_DAPI_CUS_CONTACT_INVALID</p>
p_ship_to_contact_fnrst_name	IN	VARCHAR2		<p>The first name of contact for the bill-to customer.</p> <p>Default: Defaulted from bill-to customer site level, then customer level, if defined. Otherwise, null.</p> <p>Validation: This field is mandatory.</p> <p>Error: AR_DAPI_BILL_CONTACT_NAME_INV AR_DAPI_CUS_CONTACT_INVALID</p>
p_ship_to_contact_last_name	IN	VARCHAR2		<p>The last name of contact for the bill-to customer.</p> <p>Default: Defaulted from bill-to customer site level, then customer level, if defined. Otherwise, null.</p> <p>Validation: This field is mandatory.</p> <p>Error: AR_DAPI_BILL_CONTACT_NAME_INV AR_DAPI_CUS_CONTACT_INVALID</p>

Parameter	Type	Data-type	Required*	Description
p_term_id	IN	NUMBER		<p>Payment terms identifier for the transactions. You can override payment terms.</p> <p>Default: Following hierarchy is used to default payment terms:</p> <ol style="list-style-type: none"> 1. Customer bill-to site level 2. Customer address level 3. Customer level transaction type <p>Validation: It should be a valid payment term.</p> <p>Error: AR_DAPI_TERM_NAME_INVALID AR_DAPI_TERM_ID_INVALID</p>
p_term_name	IN	VARCHAR2		<p>Payment terms name for the transactions. You can override payment terms.</p> <p>Default: Following hierarchy is used to default payment terms name:</p> <ol style="list-style-type: none"> 1. Customer bill-to site level 2. Customer address level 3. Customer level transaction type <p>Validation: It should be a valid payment term.</p> <p>Error: AR_DAPI_TERM_NAME_INVALID AR_DAPI_TERM_ID_INVALID</p>

Parameter	Type	Data-type	Required*	Description
p_salesrep_id	IN	NUMBER		<p>Salesperson identifier for the transactions. You can override salesperson.</p> <p>Default: Default the primary ID from the bill-to customer. If salescredits are required and no ID is defaulted from the bill-to customer, then p_salesrep_id is set to -3, which means "No sales credit".</p> <p>Validation: It should be a valid salesperson in the system.</p> <p>Error: AR_DAPI_SALESREP_NAME_INVALID AR_DAPI_SALESREP_ID_INVALID</p>
p_salesrep_name	IN	VARCHAR2		<p>Salesperson name for the transactions. You can override salesperson.</p> <p>Default: Default the primary from the bill-to customer. If salescredits are required and no salesperson is defaulted from the bill-to customer, then p_salesrep_name is set to -3, which means "No sales credit".</p> <p>Validation: It should be a valid salesperson in the system.</p> <p>Error: AR_DAPI_SALESREP_NAME_INVALID AR_DAPI_SALESREP_ID_INVALID</p>
p_interface_header_context	IN	VARCHAR2		<p>Interface header context.</p> <p>Default: Null</p> <p>Validation: Null</p> <p>Error: Null</p>
p_interface_header_attribute1 to p_interface_header_attribute15	IN	VARCHAR2		<p>Interface header attribute value</p> <p>Default: Null</p> <p>Validation: Null</p> <p>Error: Null</p>

Parameter	Type	Data-type	Required*	Description
p_attribute_category	IN	VARCHAR2		<p>Descriptive Flexfield structure defining column.</p> <p>Default: Null</p> <p>Validation: It should be a valid structure.</p> <p>Error: Null</p>
p_attribute1 to p_attribute15	IN	VARCHAR2		<p>Descriptive Flexfield segment column.</p> <p>Default: Null</p> <p>Validation: It should be a valid segment.</p> <p>Error: Validate_Desc_Flexfield</p>
p_global_attr_cust_r ec	IN	global_ attr_rec_type		<p>This is a record type that contains all the 25 global descriptive flexfield segments and one global descriptive flexfield structure defining column.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error:</p>
p_document_numbe r	IN	NUMBER		<p>Value assigned to document receipt.</p> <p>Default: Null.</p> <p>Validation: User should not pass the value if the current document sequence is automatic. Document sequence value should not be entered if the Sequential Numbering profile option is set to Not Used.</p> <p>Error: AR_RAPI_DOC_SEQ_AUTOMATIC AR_RAPI_DOC_SEQ_NOT_EXIST_A AR_RAPI_DOC_SEQ_NOT_EXIST_P</p>
p_ussl_transaction_ code	IN	VARCHAR2		<p>Code defined by public sector accounting.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>

Parameter	Type	Data-type	Required*	Description
p_printing_option	IN	VARCHAR2		<p>Printing option for the invoice.</p> <p>Default: Default is print option of transaction type.</p> <p>Validation: Can be 'PRI' or 'NOT'</p> <p>Error: AR_DAPI_PO_INVALID</p>
p_default_tax_exem	IN	VARCHAR2		<p>Tax exempt flag. You can enter value for the field only if the TAX: Allow Override of Customer Exception profile option is yes.</p> <p>Default: 'S' i.e. Standard</p> <p>Validation: From lookup table for lookup_type = 'TAX_CONTROL_FLAG'</p> <p>Error: AR_DAPI_STATUS_TRX_INVALID</p>
p_status_trx	IN	VARCHAR2		<p>Status of the transaction. This is a user-maintainable field and it can be defined in lookup table.</p> <p>Default: OP, can be CL, PEN, VD</p> <p>Validation: from lookup table for LOOKUP_TYPE = 'INVOICE_TRX_STATUS'</p> <p>Error: AR_DAPI_STATUS_TRX_INVALID</p>
p_financial_charges	IN	VARCHAR2		<p>Indicates whether financial charges are calculated.</p> <p>Default: Null</p> <p>Validation: can be null, Y, N</p> <p>Error: AR_DAPI_FC_INVALID</p>
p_agreement_id	IN	NUMBER		<p>Agreement associated with transaction for the customer.</p> <p>Default: Null</p> <p>Validation: Null</p> <p>Error: Null</p>

Parameter	Type	Data-type	Required*	Description
p_special_instructions	IN	VARCHAR2		<p>Any special instruction for the transaction, up to 240 characters.</p> <p>Default: Null</p> <p>Validation: Null</p> <p>Error: Null</p>
p_comments				User's comments.
p_purchase_order	IN	VARCHAR2		<p>Purchase order number.</p> <p>Default: Null</p> <p>Validation: Null</p> <p>Error: Null</p>
p_purchase_order_revision	IN	VARCHAR2		<p>Purchase order revision number.</p> <p>Default: Null</p> <p>Validation: Null</p> <p>Error: Null</p>
p_purchase_order_date	IN	DATE		<p>Purchase order date.</p> <p>Default: Null</p> <p>Validation: Null</p> <p>Error: Null</p>
p_remit_to_address_id	IN	NUMBER		<p>Remit-to address ID for the customer</p> <p>Default: Remit_to_address assigned to country, state, and postal code combination for the customer's address.</p> <p>Validate from the view: AR_ACTIVE_REMIT_TO_ADDRESSES_V</p> <p>Error: AR_DAPI_LOC_SITE_NUM_IGN AR_DAPI_REMIT_ADDR_ID_INVD</p>

Parameter	Type	Data-type	Required*	Description
p_sold_to_customer_id	IN	NUMBER		<p>The customer_id for the sold-to customer. Default: Bill_to_customer_id</p> <p>Validation:</p> <ul style="list-style-type: none"> Customer exists and has prospect code = CUSTOMER Customer has a profile defined at customer level Either bill-to or ship-to customer must exist <p>Error:</p> <p>AR_DAPI_SOLD_CUST_COM_INVALID AR_DAPI_SOLD_CUS_IGN AR_DAPI_SOLD_CUST_ID_INVALID</p>
p_sold_to_customer_name	IN	VARCHAR2		<p>The name for the entered/defaulted sold-to customer. Default: none</p> <p>Validation:</p> <ol style="list-style-type: none"> Customer exists and has prospect code = CUSTOMER Customer has a profile defined at customer level Either bill-to or ship-to customer must exist <p>Error:</p> <p>AR_DAPI_SOLD_CUST_NAME_INVALID AR_DAPI_SOLD_CUST_COM_INVALID</p>

Parameter	Type	Data-type	Required*	Description
p_sold_to_customer_number	IN	VARCHAR2		<p>The number for the entered/defaulted sold-to customer.</p> <p>Default: None</p> <p>Validation: Customer exists and has prospect code = CUSTOMER. Customer has a profile defined at customer level. Either bill-to or ship-to customer must exist.</p> <p>Error: <code>AR_DAPI_SOLD_CUST_NUM_INVALID</code> <code>AR_DAPI_SOLD_CUST_COM_INVALID</code></p>
p_paying_customer_id				<p>The customer_id associated with the customer bank account assigned to your transaction.</p> <p>Default: Same as bill-to customer</p> <p>Validation: Customer exists and has prospect code = CUSTOMER. Customer has a profile defined at customer level. Either bill-to or ship-to customer must exist.</p> <p>Error: <code>AR_DAPI_CUS_NAME_NUM_IGN</code> <code>AR_DAPI_PAY_CUST_ID_INVALID</code></p>
p_paying_customer_name				<p>The name for the entered/defaulted paying customer.</p> <p>Default: None</p> <p>Validation: Customer exists and has prospect code = CUSTOMER. Customer has a profile defined at customer level. Either bill-to or ship-to customer must exist.</p> <p>Error: <code>AR_DAPI_PAY_CUST_NAME_INVALID</code> <code>AR_DAPI_PAY_CUST_COM_INVALID</code></p>

Parameter	Type	Data-type	Required*	Description
p_paying_customer_number				<p>The number for the entered/defaulted paying customer.</p> <p>Default: None</p> <p>Validation: Customer exists and has prospect code = CUSTOMER. Customer has a profile defined at customer level. Either bill-to or ship-to customer must exist.</p> <p>Error: AR_DAPI_PAY_CUST_NUM_INVALID AR_DAPI_PAY_CUST_COM_INVALID</p>
p_paying_location				<p>The location for the paying customer.</p> <p>Default: Null</p> <p>Validation: This field is mandatory.</p> <p>Error: AR_DAPI_CUS_LOC_INVALID</p>
p_receipt_method_i_d	IN	NUMBER		<p>Identifies the receipt method of the transactions.</p> <p>Default: From receipt method name.</p> <p>Validation: Validation detailed in Example, page 5-25.</p> <p>Error: AR_RAPI_RCPT_MD_NAME_IGN AR_RAPI_RCPT_MD_ID_INVALID</p>
p_receipt_method_name	IN	VARCHAR2		<p>The receipt method name of the transactions.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPI_RCPT_MD_NAME_INVALID</p>

Parameter	Type	Data-type	Required*	Description
p_cust_bank_acco nt_id	IN	NUMBER		<p>Customer bank account identifier.</p> <p>Default: None</p> <p>Validation: From AP_BANK_ACCOUNTS table.</p> <p>Error: AR_RAPI_CUS_BK_NAME_NUM_IGN AR_RAPI_CUS_BK_AC_ID_INVALID</p>
p_cust_bank_acco nt_name	IN	VARCHAR2		<p>Customer bank account name.</p> <p>Default: None</p> <p>Validation: From AP_BANK_ACCOUNTS table.</p> <p>Error: AR_RAPI_CUS_BK_AC_NAME_INVALID AR_RAPI_CUS_BK_AC_2_INVALID</p>
p_cust_bank_acco nt_number	IN	VARCHAR2		<p>Customer bank account number.</p> <p>Default: None</p> <p>Validation: From AP_BANK_ACCOUNTS table.</p> <p>Error: AR_RAPI_CUS_BK_AC_NUM_INVALID AR_RAPI_CUS_BK_AC_2_INVALID</p>
p_start_date_commi tment	IN	DATE		<p>Start date of commitment.</p> <p>Default: Sysdate</p> <p>Validation: Based on end date, etc.</p> <p>Error: AR_TW_BAD_COMMITMT_DATE_RAN GE AR_TW_COMMIT_END_TRX_DATE AR_TW_BAD_DATE_COMMITMENT</p>

Parameter	Type	Data-type	Required*	Description
p_end_date_commitment	IN	DATE		<p>End date of commitment.</p> <p>Default: Null</p> <p>Validation: Based on start date, etc.</p> <p>Error: AR_TW_BAD_COMMITMT_DATE_RAN GE AR_TW_COMMIT_END_TRX_DATE AR_TW_BAD_DATE_COMMITMENT</p>
p_amount	IN	NUMBER		<p>Deposit amount.</p> <p>Default: Cannot be negative.</p> <p>Validation: Based on start date, etc. This field is mandatory.</p> <p>Error: AR_DAPI_COMM_AMOUNT_NULL AR_TW_COMMIT_AMOUNT_NEGATIVE</p>
p_inventory_id	IN	NUMBER		<p>Item ID of commitment. You can enter memo or item ID.</p> <p>Default: Null</p> <p>Validation: Based on MTL_SYSTEM_ITEMS_B table.</p> <p>Error: AR_DAPI_INV_ID_INVALID AR_DAPI_INV_MEMO_COM</p>
p_memo_line_id	IN	NUMBER		<p>Memo line ID. You can enter memo or item ID.</p> <p>Default: Null</p> <p>Validation: Based on AR_MEMO_LINES table.</p> <p>Error: AR_DAPI_MEMO_NAME_INVALID AR_DAPI_MEMO_WRG AR_DAPI_INV_MEMO_COM</p>

Parameter	Type	Data-type	Required*	Description
p_memo_line_name	IN	VARCHAR2		<p>Deposit amount.</p> <p>Default: Null</p> <p>Validation: Based on AR_MEMO_LINES table.</p> <p>Error: AR_DAPI_MEMO_NAME_INVALID AR_DAPI_MEMO_WRG</p>
p_description	IN	VARCHAR2		<p>Description of deposit.</p> <p>Default: Null</p> <p>Validation: Null</p> <p>Error: Null</p>
p_comm_interface_li_ne_context	IN	VARCHAR2		<p>Interface line context for deposit.</p> <p>Default: Null</p> <p>Validation: Null</p> <p>Error: Null</p>
p_comm_interface_li_ne_attr1 to p_comm_interface_li_ne_attr15	IN	VARCHAR2	NULL	<p>Interface line attribute value for deposit.</p> <p>Default: Null</p> <p>Validation: Null</p> <p>Error: Null</p>
p_comm_attr_catego ry	IN	VARCHAR2	NULL	<p>Descriptive Flexfield structure defining column for deposit lines.</p> <p>Default: Null</p> <p>Validation: It should be a valid structure.</p> <p>Error: Null</p>

Parameter	Type	Data-type	Required*	Description
p_comm_attr1 to p_comm_attr15	IN	VARCHAR2	NULL	<p>Descriptive Flexfield segment column for deposit lines.</p> <p>Default: Null</p> <p>Validation: It should be a valid segment.</p> <p>Error: Validate_Desc_Flexfield</p>
p_global_attr_cust_lines_rec	IN	global_attr_rec_type	NULL	<p>This is a record type that contains all the 25 global descriptive flexfield segments for deposit lines and one global descriptive flexfield structure defining column.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_owner_id	IN	NUMBER	Null	<p>ID of the commitment owner.</p> <p>Default: None</p> <p>Validation: Yes (same as customer contact).</p> <p>Error: N/A</p>
p_owners_name	IN	NUMBER	Null	<p>Name of the commitment owner.</p> <p>Default: None</p> <p>Validation: Yes (same as customer contact)</p> <p>Error: N/A</p>
X_new_trx_number	OUT	VARCHAR2		New transaction number, if generated.
X_new_customer_trx_id	OUT	VARCHAR2		New CUSTOMER_TRX_ID of the deposit created.
X_new_customer_trx_line_id	OUT	VARCHAR2		New CUSTOMER_TRX_LINE_ID of the deposit created.
X_new_rowid	OUT	VARCHAR2		Row ID of the deposit created.
X_new_status	OUT	VARCHAR2		Status of the deposit created.

Example

Objective:

To create a deposit using a call to ar_deposit_api_pub.Create_deposit and passing a minimum number of Input parameters.

Entered parameters:

```
p_api_version      =1.0  ,
p_init_msg_list    ='F'  ,
p_deposit_number   = 'Your Deposit Number'
p_deposit_date     = sysdate,
p_currency_code    ='USD',
p_batch_source_id  = Choose a Valid Batch source ID
p_cust_trx_type_id = Choose a Valid Transaction Type ID of class
"Deposit"
p_class            ='DEP' i.e. Depsoit
p_bill_to_customer_number = Choose a Valid Customer Number
p_start_date_commitment = sysdate
p_amount           = Choose deposit Amount
p_description      = Your Deposit Description
```

Before calling the APIs you should set up the application, responsibility and the user in the context of Oracle Applications by calling the following FND API.

```
fnd_global.apps_initialize ( user_id =>'Your user id', resp_id => 'Your
Responsibility id', resp_appl_id => 'Your Application id');
```

For example:

```
fnd_global.apps_initialize ( user_id => 1318, resp_id => 50559,
resp_appl_id => 222);
```

The API call in this case would be:

```
DECLARE
l_return_status  VARCHAR2(1);
l_msg_count      NUMBER;
l_msg_data        VARCHAR2(240);
l_count           NUMBER;
l_new_trx_number    ra_customer_trx.trx_number%type;
l_new_customer_trx_id ra_customer_trx.customer_trx_id%type;
l_new_customer_trx_line_id
ra_customer_trx_lines.customer_trx_line_id%type;
l_new_rowid        VARCHAR2(240);
l_new_status       VARCHAR2(240);

BEGIN
fnd_global.apps_initialize ( user_id => 1318, resp_id => 50559,
resp_appl_id => 222);
ar_deposit_api_pub.CREATE_DEPOSIT(
```

1. Standard API parameters.

```

p_api_version          => 1.0,
p_init_msg_list        => FND_API.G_TRUE,
p_commit               => FND_API.G_TRUE,
p_validation_level    => FND_API.G_VALID_LEVEL_FULL,
x_return_status         => l_return_status,
x_msg_count            => l_msg_count,
x_msg_data             => l_msg_data,
p_deposit_number       => 'dapi'||userenv('SESSIONID'),
p_deposit_date         => sysdate,
p_currency_code        => 'USD',
p_batch_source_id     => 'Choose a Valid Batch source ID',
p_cust_trx_type_id   => 'Choose a Valid Transaction Type ID of class
Deposit',
p_class                => 'DEP' ,
p_bill_to_customer_number => 'Choose a Valid Customer Number',
p_start_date_commitment  => sysdate,
p_amount               => 'Choose deposit Amount',
p_description          => 'Your Deposit Description',
X_new_trx_number       =>l_new_trx_number,
X_new_customer_trx_id  =>l_new_customer_trx_id,
X_new_customer_trx_line_id =>l_new_customer_trx_line_id,
X_new_rowid             =>l_new_rowid,
X_new_status            =>l_new_status ) ;
IF l_msg_count  = 1 Then

```

2. There is one message raised by the API, so it has been sent out.
3. In the parameter x_msg_data, get it.
4. The messages on the stack are more than one, so call them in a loop.
5. And print the messages.

```

dbms_output.put_line('l_msg_data'||l_msg_data);
ELSIF l_msg_count  > 1 Then

```

```

LOOP
  IF nvl(l_count,0) < l_msg_count THEN
    l_count := nvl(l_count,0) +1 ;
    l_msg_data := FND_MSG_PUB.Get(FND_MSG_PUB.G_NEXT,FND_API.G_FALSE);
      IF l_count = 1 THEN
        dbms_output.put_line('l_msg_data 1'||l_msg_data);
      ELSIF l_count = 2 THEN
        dbms_output.put_line('l_msg_data 2'||l_msg_data);
      ELSIF l_count = 3 THEN
        dbms_output.put_line('l_msg_data 3'||l_msg_data);
      ELSIF l_count = 4 THEN
        dbms_output.put_line('l_msg_data 4'||l_msg_data);
      ELSIF l_count = 5 THEN
        dbms_output.put_line('l_msg_data 5'||l_msg_data);
      ELSIF l_count = 6 THEN
        dbms_output.put_line('l_msg_data 6'||l_msg_data);
      END IF;
      dbms_output.put_line('l_msg_data
'||to_char(l_count)||':'||l_msg_data);
    ELSE
      EXIT;
    END IF;
  END LOOP;
END IF;
Commit;
END;

```

Depending on the message level threshold set the profile option FND_API_MSG_LEVEL_THRESHOLD, the messages put on the message stack may contain both the error messages and the warnings.

AR_DEPOSIT_API_PUB.insert_non_rev_salescredit

This routine is called to assign nonrevenue sales credit to salespersons for a deposit. You can create as many of the nonrevenue credit assignments as you need.

This API routine has 4 output and 22 input parameters in total.

The following is the breakdown of the parameters:

Input

Standard API parameters: 4

Owners parameters: 22

Output

Standard API parameters: 3

Owners parameters: 0

Parameter Descriptions

The following table lists the API parameters.

Parameter	Type	Data-type	Required	Default Value	Description
p_api_version	IN	NUMBER	Yes		Used to compare version numbers of incoming calls to its current version number. Unexpected error is raised if version incompatibility exists. In the current version of the API, you should pass in a value of 1.0 for this parameter.
p_init_msg_list	IN	VARCHAR 2		FND_API.G_FALSE	Allows API callers to request that the API does initialization of the message list on their behalf.
p_commit	IN	VARCHAR 2		FND_API.G_FALSE	Used by API callers to ask the API to commit on their behalf.
p_validation_level	IN	NUMBER		FND_API.G_VALID_LEVEL_FULL	Not to be used currently as this is a public API.
x_return_status	OUT	VARCHAR 2			Represents the API overall return status.
x_msg_count	OUT	NUMBER			Number of messages in the API message list.
x_msg_data	OUT	VARCHAR 2			This is the message in encoded format if x_msg_count=1.

The following table lists the parameters relevant to the deposit.

Parameter	Type	Data-type	Required	Description
p_deposit_number	IN	VARCHAR 2	Null	Deposit number, same as trx_number for the transaction number. Default: None Validation: Yes Error: N/A

Parameter	Type	Data-type	Required	Description
p_customer_trx_id	IN	NUMBER		<p>Customer_trx_id of the deposit created.</p> <p>Default: None</p> <p>Validation: Yes</p> <p>Error: N/A</p>
p_salesrep_number	IN	NUMBER	Null	<p>Salesperson number.</p> <p>Default: None</p> <p>Validation: Yes (same as customer contact).</p> <p>Error: N/A</p>
p_salesrep_id	IN	NUMBER		<p>Salesrep_id of the salesperson.</p> <p>Default: None</p> <p>Validation: Yes</p> <p>Error: N/A</p>
p_non_revenue_amo unt_split	IN	NUMBER		<p>Nonrevenue credit amount associated with salesperson.</p> <p>Default: None</p> <p>Validation: Yes</p> <p>Error: N/A</p>
p_non_revenue_perc ent_split	IN	NUMBER		<p>Nonrevenue credit percent associated with salesperson.</p> <p>Default: None</p> <p>Validation: Yes</p> <p>Error: N/A</p>
p_attribute_category	IN	VARCHAR 2		<p>Descriptive Flexfield structure defining column.</p> <p>Default: Null</p> <p>Validation: It should be a valid structure.</p> <p>Error: Null</p>

Parameter	Type	Data-type	Required	Description
p_attribute1 to p_attribute15	IN	VARCHAR2 2		<p>Descriptive Flexfield segment column. Default: Null</p> <p>Validation: It should be a valid segment. Error: Validate_Desc_Flexfield</p>

Example

Objective:

To create owner assignment using ar_deposit_api_pub.insert_non_rev_salescredit and passing a minimum number of Input parameters.

Entered parameters:

```
p_api_version      => 1.0 ,
p_init_msg_list   => 'F',
,p_customer_trx_id => 'Valid Customer Trx ID, Must be a deposit'
,p_salesrep_id     => -3 , means no Sales Rep
p_non_revenue_percent_split => 300
```

Before calling the APIs you should set up the application, responsibility and the user in the context of Oracle Application by calling the following FND API.

```
fnd_global.apps_initialize ( user_id =>'Your user id', resp_id => 'Your
Responsibility id', resp_appl_id => 'Your Application id');
```

For example:

```
fnd_global.apps_initialize ( user_id => 1318, resp_id => 50559,
resp_appl_id => 222);
```

The API call in this case would be:

```
DECLARE
l_return_status  VARCHAR2(1);
l_msg_count      NUMBER;
l_msg_data        VARCHAR2(240);
l_count           NUMBER;

BEGIN
fnd_global.apps_initialize ( user_id => 1318, resp_id => 50559,
resp_appl_id => 222);
ar_deposit_api_pub.insert_non_rev_salescredit()
```

1. Standard API parameters.

```

p_api_version      => 1.0,
p_init_msg_list    => FND_API.G_TRUE,
p_commit           => FND_API.G_TRUE,
p_validation_level => FND_API.G_VALID_LEVEL_FULL,
x_return_status     => l_return_status,
x_msg_count         => l_msg_count,
x_msg_data          => l_msg_data,
p_customer_trx_id  => ' Valid Customer Trx ID ,
Must be a deposit',
p_salesrep_id       => -3,
p_non_revenue_amount_split => 300);

dbms_output.put_line('return status '||l_return_status);
dbms_output.put_line('l_msg_count
'||to_char(l_msg_count));

IF l_msg_count  = 1 Then
  2. There is one message raised by the API, so it has been sent out.

  3. In the parameter x_msg_data, get it.

  dbms_output.put_line('l_msg_data '||l_msg_data);
  ELSIF l_msg_count  > 1 Then
    4. The messages on the stack are more than one, so call them in a loop.

    5. And print the messages.

  LOOP
    IF nvl(l_count,0) < l_msg_count THEN
      l_count := nvl(l_count,0) +1 ;
      l_msg_data := FND_MSG_PUB.Get(FND_MSG_PUB.G_NEXT,FND_API.G_FALSE);
        IF l_count = 1 THEN
          dbms_output.put_line('l_msg_data 1 '||l_msg_data);
        ELSIF l_count = 2 THEN
          dbms_output.put_line('l_msg_data 2 '||l_msg_data);
        ELSIF l_count = 3 THEN
          dbms_output.put_line('l_msg_data 3 '||l_msg_data);
        ELSIF l_count = 4 THEN
          dbms_output.put_line('l_msg_data 4 '||l_msg_data);
        ELSIF l_count = 5 THEN
          dbms_output.put_line('l_msg_data 5 '||l_msg_data);
        ELSIF l_count = 6 THEN
          dbms_output.put_line('l_msg_data 6 '||l_msg_data);
        END IF;
        dbms_output.put_line('l_msg_data
'||to_char(l_count)||': '||l_msg_data);
      ELSE
        EXIT;
      END IF;
    END LOOP;
  END IF;

  Commit;
END;

```

Depending on the message level threshold set the profile option FND_API_MSG_LEVEL_THRESHOLD, the messages put on the message stack may contain both the error messages and the warnings.

Messages

Messages play an important role in the effectiveness of your API calls. The right message is raised at the right point to convey to you the exact error that has occurred or any warnings that have been raised.

In the Commitment (Deposit) API, all error messages and warnings raised during the execution are put on the message stack and can be retrieved by the user as described in Exception Handling and Result Messages, page 1-3.

WARNINGS AND ERRORS

The following table contains the list of all the error messages raised by the Commitment (Deposit) API.

Message Number	Message Code	Message Text	Type
294849	AR_DAPI_COMM_AMOUNT_NULL	The commitment amount requires a value.	E
294850	AR_DAPI_CUS_LOC_INVALID	The customer location is invalid.	E
294851	AR_DAPI_CUS_SITE_DFT_INVALID	The customer site use ID could not be defaulted.	E
294852	AR_DAPI_CUS_CONTACT_INVALID	The customer contact is invalid.	E
294853	AR_DAPI_CUST_NULL	A value for the customer ID, name, or number is required.	E
294854	AR_DAPI_COMM_BATCH_INVALID	The batch name or ID is invalid.	E
294855	AR_DAPI_TRANS_TYPE_ID_INVALID	The transaction type ID is invalid.	E
294856	AR_DAPI_TRANS_TYPE_INVALID	The transaction type is invalid.	E
294857	AR_DAPI_TERM_NAME_INVALID	The term name is invalid.	E
294858	AR_DAPI_TERM_ID_INVALID	The term ID is invalid.	E
294859	AR_DAPI_SALESREP_NAME_INVALID	The sales representative name is invalid.	E
294860	AR_DAPI_SALESREP_ID_INVALID	The sales representative ID is invalid.	E

Message Number	Message Code	Message Text	Type
294861	AR_DAPI_BS_NAME_INVALID	The batch source name is invalid.	E
294862	AR_DAPI_BS_ID_INVALID	The batch source ID is invalid.	E
	AR_DAPI_BS_NAME_IGN	The batch source name has been ignored.	W
294863	AR_DAPI_SOLD_CUST_NAME_INVALID	The sold-to customer name is invalid.	E
	AR_DAPI_SOLD_CUST_DFT	The sold-to customer defaulted to the bill-to customer.	W
294864	AR_DAPI_SOLD_CUST_COM_INVALID	The combination of sold-to customer name and number must be valid.	E
294865	AR_DAPI_PAY_CUST_NAME_INVALID	The paying customer name is invalid.	E
	AR_DAPI_PAY_CUST_DFT	The paying customer defaulted to the bill-to customer.	W
294866	AR_DAPI_PAY_CUST_COM_INVALID	The combination of paying customer name and number must be valid.	E
294867	AR_DAPI_PAY_CUST_NUM_INVALID	The paying customer number is invalid.	E
	AR_DAPI_CUS_NAME_NUM_IGN	The paying customer name and number have been ignored.	W
294868	AR_DAPI_PAY_CUST_ID_INVALID	The paying customer ID is invalid.	E
294869	AR_DAPI_SOLD_CUST_ID_INVALID	The sold-to customer ID is invalid.	E
	AR_DAPI_SOLD_CUS_IGN	The sold-to customer name and number have been ignored.	W
	AR_DAPI_PO_INVALID	The printing option is invalid.	E
294871	AR_DAPI_STATUS_TRX_INVALID	The transaction status is invalid.	E
294872	AR_DAPI_TAX_FLAG_INVALID	The default tax flag is invalid.	E
	AR_DAPI_NO_BATCH	A batch or a batch in the profile is required.	E

Message Number	Message Code	Message Text	Type
294874	AR_DAPI_MEMO_NAME_INVALID	The memo name is invalid.	E
	AR_DAPI_MEMO_WRG	The memo ID, not the provided memo name, has been used.	W
	AR_DAPI_TRANS_TYPE_IGN	The type ID, not the provided type, has been used.	W
	AR_DAPI_INV_ID_INVALID	The inventory item ID is invalid.	E
	AR_DAPI_INV_MEMO_COM	Enter either a memo or inventory item ID.	E
294877	AR_DAPI_BILL_OR_SHIP_CUST_REQ	A bill-to or ship-to customer is required.	E
294878	AR_DAPI_BILL_CONTACT_NAME_IN_V	Both a first and last name are required for the bill-to contact.	E
294879	AR_DAPI_SHIP_CONTACT_NAME_IN_V	Both a first and last name are required for the ship-to contact.	E
	AR_DAPI_DEPOSIT_NO_NULL	A deposit number is required.	E
294881	AR_DAPI_FC_INVALID	The finance charges are invalid.	E
	AR_DAPI_LOC_SITE_NUM_IGN	The location site number has been ignored.	W
294882	AR_DAPI_REMIT_ADDR_ID_INVD	The remit-to address ID is invalid.	E
294883	AR_DAPI_CUST_LOC_SITE_NUM_INV	The customer location site number is invalid.	E
294884	AR_DAPI_REMIT_ADDRESS_DFT_ERR	The remit-to address did not successfully default.	E
294885	AR_DAPI_TRANS_TYPE_NULL	A value for either the transaction type or ID is required.	E
294886	AR_DAPI_BILL_CONTACT_COM_INV	The combination of the bill-to contact's first and last name must be valid.	E

Message Number	Message Code	Message Text	Type
294887	AR_DAPI_SHIP_CONTACT_COM_INV	The combination of the ship-to contact's first and last name must be valid.	E
294888	AR_DAPI_POST_COMMIT_ST	The deposit did not successfully post.	E
294889	AR_DAPI_INSERT_HEADER_ST	The header was not successfully inserted for the deposit.	E
	AR_DAPI_BILL_VAL_SHIP_IGN	The bill-to customer was defaulted from the ship-to customer because a value for the bill-to customer did not exist.	W
294890	AR_DAPI_LOC_INV	The location is invalid.	E
294891	AR_DAPI_SALESREP_ST	The salesperson was not successfully inserted for the deposit.	E
294892	AR_DAPI_SALESREP_NO_ID_NAME	The salesperson ID and name are required.	E
294893	AR_DAPI_NON_REV_AMT_PCT	A percentage or amount of nonrevenue sales credit is required.	E
294894	AR_DAPI_DEP_NO_ID_REQ	A deposit number or customer transaction ID is required.	
	AR_DAPI_DEP_NO_ING	The deposit number has been ignored.	W
294895	AR_DAPI_DEP_ID_INVALID	The customer transaction ID is invalid.	E
294896	AR_DAPI_DEP_NO_INVALID	The deposit number is invalid.	E
	AR_DAPI_REV_AMT_IGN	The nonrevenue sales credit amount has been ignored.	W

Invoice Creation API User Notes

Overview

This document outlines the use of Invoice Creation API. This API allows users to create an invoice using simple calls to PL/SQL functions.

The Invoice Creation API is not intended to replace the existing Transaction workbench, AutoInvoice, or the Transaction API program.

You can access this API in two ways:

- As standard PL/SQL servers-side routine calls
- Through Forms, utilizing the capability of Forms6 to have a procedure as its underlying base table

Modular Approach

To modularize the Invoice Creation API, the basic structure of the API is divided into four parts:

1. Get all the default values from profiles and AR_SYSTEM_PARAMETERS table.
2. Populate four global temporary tables for Header, Lines, Distributions and Sales Credits from PL/SQL tables and Default values (if user has not entered).
3. Validate all the parameters entered by the user.
4. Call the entity handlers to perform the relevant task (such as Create).

This results in easy to understand and easy to maintain code. Any new functionality can be added by a simple code plug-in at each of the four parts.

Debug Messages

The Invoice Creation API uses the Oracle Applications Logging Framework to log all debug messages in a central repository. Please query using module name, ar.plsql.InvoiceAPI.

The debugging can be enabled by the setting the following profile options:

1. FND: Debug Log Enabled (AFLOG_ENABLED) to 'Y'.
2. FND: Debug Log Level (AFLOG_LEVEL) to 'Statement'.

Once the above parameters are set, the message will be logged in the FND repository. The API to log accepts log level, module name, and the actual text.

An example is given below:

```
FND_LOG.STRING (P_LOG_LEVEL, P_MODULE_NAME, P_MESSAGE);
```

All Invoice Creation API debug messages use a module name of 'ar.plsql.InvoiceAPI'.

See: *Oracle Applications Supportability Guide*.

API Usage

To create an invoice, you can call the following PL/SQL APIs:

- AR_INVOICE_API_PUB.CREATE_INVOICE: Creates multiple invoices in a batch.
- AR_INVOICE_API_PUB.CREATE_SINGLE_INVOICE: Create a single invoice and return customer_trx_id.

See: AR_INVOICE_API_PUB, page 6-2.

AR_INVOICE_API_PUB

The API contains 2 public procedures to create either a single invoice, or multiple invoices in batch mode. The input parameters are the same for both procedures and are explained in the following section.

- Use the CREATE_INVOICE procedure to create multiple invoices in a batch. The procedure returns a global record type structure which contains the batch_id to retrieve the necessary data from the transaction tables. The structure is defined in the package specification of ar_invoice_api_pub. Please refer to Example for Creating Multiple Invoices in a Batch, page 6-18 for usage.

```
TYPE api_outputs_type IS RECORD
(
  batch_id NUMBER DEFAULT NULL
);
```

- Use the CREATE_SINGLE_INVOICE procedure to create a single invoice. The procedure returns customer_trx_id as an out parameter, but the procedure does not create a batch for the single invoice. Please refer to Example for Creating a Single Invoice, page 6-21 for usage.

API Parameters

The API accepts the following parameters:

p_api_version	IN	NUMBER		
p_init_msg_list	IN	VARCHAR2	:= FND_API.G_FALSE,	
p_commit	IN	VARCHAR2	:= FND_API.G_FALSE,	
p_batch_source_rec	IN	batch_source_rec_type,		
p_trx_header_tbl	IN	trx_header_tbl_type,		
p_trx_lines_tbl	IN	trx_line_tbl_type,		
p_trx_dist_tbl	IN	trx_dist_tbl_type,		
p_trx_salescredits_tbl	IN	trx_salescredits_tbl_type,		
p_trx_contingencies_tbl	IN	trx_contingencies_tbl_type		
x_customer_trx_id	OUT NOCOPY	NUMBER,		
x_return_status	OUT NOCOPY	VARCHAR2,		
x_msg_count	OUT NOCOPY	NUMBER,		
x_msg_data	OUT NOCOPY	VARCHAR2,		

The following table shows the list of standard API parameters.

Parameter	Type	Data Type	Required	Default Value	Description
p_api_version	IN	NUMBER	Yes	1.0	Compare version numbers of incoming calls to its current versions
p_init_msg_list	IN	VARCHAR2		FND_API.G_FALSE	Allow API callers to request that API does initialize the message list on their behalf.
p_commit	IN	VARCHAR2		FND_API.G_FALSE	Used by API callers to ask the API to commit on their behalf.
x_customer_trx_id	OUT	NUMBER			Returns customer_trx_id in case it is called for creating a single invoice. This parameter works only with CREATE_SINGLE_INVOICE procedure.
x_return_status	OUT	VARCHAR2			Represent the API status.

Parameter	Type	Data Type	Required	Default Value	Description
x_msg_count	OUT	NUMBER			Number of messages in the PI message list (not used by this API).
x_message_data	OUT	VARCHAR2			Message in case API encounters any unexpected error.

P_BATCH_SOURCE_REC Parameter

The P_BATCH_SOURCE_REC parameter is of PL/SQL record type, and has the following attributes, as described in this table:

Attribute Name	Data Type	Required	Default Value	Description
batch_source_id	NUMBER		Null	If batch_source_id is null then value will be derived from AR_RA_BATCH_SOURCE profile option. In case the value is passed then it will be validated against ra_batch_sources. Only 'Manual' batch sources are allowed.
default_date	DATE		Null	If the value is null then Sysdate will be taken.

P_TRX_CONTINGENCIES_TBL Parameter

The P_TRX_CONTINGENCIES_TBL parameter is of PL/SQL table type TRX_CONTINGENCIES_REC_TYPE.

TRX_CONTINGENCIES_REC_TYPE has the following attributes, as described in this table:

Attribute Name	Data Type	Required	Default Value	Description
trx_contingency_id	NUMBER			Unique identifier for each contingency on a line.
trx_line_id	NUMBER			Identifies rows in the user defined line table.
contingency_code	VARCHAR2(30)			Obsolete.

Attribute Name	Data Type	Required	Default Value	Description
contingency_id	NUMBER			Identifies the actual revenue contingency being requested. Valid values are maintained in the AR_DEFERRAL_REASON table.
expiration_event_date	DATE		NULL	Indicates the expiration of the contingency removal event. Normally defaulted by the API, the expiration_event_date is set as one of the following based on settings in AR_DEFERRAL_REASON: transaction date, fulfillment date, ship confirm date, or proof of delivery date. However, you can override the default date.
expiration_days	NUMBER		NULL	The offset days that are added to the expiration_event_date to calculate the actual expiration_date.
expiration_date	DATE		NULL	The date on which a contingency is scheduled to expire (unless another event causes it to expire earlier). The actual expiration_date is calculated by adding the expiration_days to the expiration_event_date.
attribute_category	VARCHAR2(30)		NULL	Descriptive flexfield structure definition column.
attribute1 - 15	VARCHAR2(150)		Null	Descriptive flexfield segment.
completed_flag	VARCHAR2(1)		'N'	Determines if the contingency is active or expired at the time of creation. If set to 'Y', then the contingency does not bar or withhold revenue from being recognized.
completed_by	NUMBER		NULL	User who completed the contingency.

P_TRX_HEADER_TBL Parameter

The P_TRX_HEADER_TBL parameter is of PL/SQL table type TRX_HEADER_REC_TYPE.

TRX_HEADER_REC_TYPE has the following attributes, as described in this table:

Attribute Name	Data Type	Required	Default Value	Description
trx_header_id	NUMBER	Yes		Identifier for the Invoice header record. This must be unique for each record. This column can be generated based on a sequence or any number value. The value does not get recorded into any table.
trx_number	VARCHAR2(30)	Null		This is the transaction number for the invoice. This field should not be populated if the batch source has Copy Document Sequence Number to Transaction Number checked or if Automatic Transaction Numbering is enabled.
trx_date	DATE	Null		Invoice Date. If no value is passed then p_batch_source_rec.default_date is used. If that too is not passed then sysdate is used.
gl_date	DATE	Null		General ledger Date. If no date is passed then p_batch_source_rec.default_date is used. If that too is not passed then sysdate is used.
trx_currency	VARCHAR2(30)	Null		Transaction Currency. If not populated then ar_system_parameters is used to retrieve it. The currency if populated must be active as of the transaction date.
cust_trx_type_id	NUMBER	Null		Transaction Type Identifier. This can have any one of the following three values: 'INV', 'DM', or 'CM'. Validated against ra_cust_trx_types. If not populated, then it is retrieved from the batch source.
bill_to_customer_id	NUMBER	Yes		Bill To Customer ID. This must exist in hz_cust_accounts table. The customer must be an active ('A') customer. Validated against hz_cust_accounts.cust_account_id.
bill_to_account_number	VARCHAR2(30)	Null		Bill To Customer Number. If both Bill To Customer ID and Bill To Customer Number are passed, then the former will take precedence. Validated against hz_cust_accounts.account_number.

Attribute Name	Data Type	Required	Default Value	Description
bill_to_customer_name	VARCHAR2(260)	Null		Bill To Customer Name. If all three are passed, the precedence is as follows: Customer ID, Customer Number, then Customer Name.
bill_to_contact_id	NUMBER	Null		Bill To Customer Contact ID. This must exist for the Bill To Customer and Bill To Address combination.
bill_to_address_id	NUMBER	Null		Bill To Address ID. This must exist in hz_cust_acct_sites for the populated Bill To Customer ID
bill_to_site_use_id	NUMBER	Null		Bill To Site use ID. The site use ID must exist in combination with Ship To Customer ID, Ship To Address ID.
ship_to_customer_id	NUMBER			Ship To Customer ID. This must exist in hz_cust_accounts table.
ship_to_account_number	VARCHAR2(30)	Null		Ship To Customer Number. If both Bill To Customer ID and Ship To Customer Number are passed, then the former will take precedence.
ship_to_customer_name	VARCHAR2(260)	Null		Ship To Customer Name. If all three are passed, the precedence is as follows: Customer ID, Customer Number, then Customer Name.
ship_to_contact_id	NUMBER	Null		Ship To Customer Contact ID. This must exist for the Ship To Customer and Ship To Address combination.
ship_to_address_id	NUMBER	Null		Ship To Address ID. This must exist in hz_cust_acct_sites for the populated Ship To Customer ID.
ship_to_site_use_id	NUMBER	Null		Ship To Site use ID. The site use ID must exist in combination with Ship To Customer ID, Ship To Address ID.
sold_to_customer_id	NUMBER	Null		Ship To Customer ID. This must exist in hz_cust_accounts table.

Attribute Name	Data Type	Required	Default Value	Description
term_id	NUMBER	Null		Payment Terms Identifier. The Term ID must be valid for the transaction date. If not populated, then it is retrieved from ra_terms based on bill_to_customer_id and bill_to_site_use_id.
primary_salesrep_id	NUMBER	Null		Primary Salesrep ID. This is required if Salesperson check box is checked in the System Options form. If not populated, then it is derived based on bill-to_customer_id and bill_to_site_use_id.
primary_salesrep_name	VARCHAR2(240)	Null		Primary Salesrep name. If both salesrep ID and name are passed, then Salesrep ID will take precedence.
exchange_rate_type	VARCHAR2(60)	Null		Exchange Rate Type. This must exist in gl_daily_conversion_types. Required if trx_currency is different from functional currency. If not populated, then it will derive from gl.
exchange_date	DATE	Null		Exchange Date. Required if trx_currency is different from functional currency. If not populated, then it will derive from gl.
exchange_rate	NUMBER	Null		Exchange Rate. This should be entered only if transaction currency is different from the functional currency and exchange rate type is 'User'.
territory_id	NUMBER	Null		Territory ID. If not populated, then it is defaulted based on the following hierarchy: <ul style="list-style-type: none"> • The Bill To site use • The Ship To Site Use • The Primary Salesrep's territory depending on the value of the DEFAULT_TERRITORY system option

Attribute Name	Data Type	Required	Default Value	Description
remit_to_address_id	NUMBER	Null		Remit To Address ID. If not populated, then it is defaulted based on country, state, and postal code of bill_to_site_use_id. If populated, then validated against ar_active_remit_to_addresses_v.
invoicing_rule_id	NUMBER	Null		Invoicing Rule ID. Valid values are -2 and -3. If you enter a value here, then you must populate accounting rule for line type = 'LINE'.
printing_option	VARCHAR2(20)	Null		Revenue Accounting lookup code for INVOICE_PRINT_OPTIONS. Valid codes are PRI - Print and NOT - Do not Print.
purchase_order	VARCHAR2(50)	Null		Purchase Order Number for this transaction.
purchase_order_revision	VARCHAR2(50)	Null		Purchase Order Revision. This must not be entered if purchase order is not populated.
purchase_order_date	DATE	Null		Purchase Order date. This must not be entered if purchase order is not populated.
comments	VARCHAR2 (240)	Null		Comments. Value can be printed on an invoice using the Print Invoice view.
internal_notes	VARCHAR2 (240)	Null		Stores the special instruction. Value can be printed on an invoice using the Print Invoice view.
finance_charges	VARCHAR2(1)	Null		Indicates if finance charges are included. Y for yes, N otherwise.

Attribute Name	Data Type	Required	Default Value	Description
receipt_method_id	NUMBER	Null		<p>This is the payment identifier for this transaction. If not populated, then it is defaulted based on the following hierarchy:</p> <ol style="list-style-type: none"> 1. Primary receipt method of 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
related_customer_tx_x_id	NUMBER	Null		Customer transaction ID of the document to which this transaction is related. Validated against <code>ra_customer_trx_all.customer_trx_id</code> . Not required for on-account credit memos.
agreement_id	NUMBER	Null		Customer Agreement identifier for this transaction. If not populated, then it will be defaulted from the commitment. Must exist in <code>SO AGREEMENTS</code> . (For future use.)
ship_via	VARCHAR2(30)	Null		Ship Via Code. If populated, then validated against <code>org_freight</code> .
ship_date_actual	DATE	Null		Ship Date
waybill_number	VARCHAR2(50)	Null		Waybill Number
fob_point	VARCHAR2(30)	Null		Free on Board Point. Validated against <code>AR LOOKUPS.LOOKUP_TYPE='FOB'</code> .

Attribute Name	Data Type	Required	Default Value	Description
customer_bank_account_id	NUMBER	Null		<p>Customer bank account ID. If the receipt method is Automatic, then it is required. If not populated, then it will be default using the following hierarchy.</p> <ol style="list-style-type: none"> 1. Primary bank account assigned to the primary site. 2. Primary bank assigned to parent customer. 3. Primary bank assigned to bill to site use. 4. Primary bank assigned to bill to customer.
default_ussgl_transaction_code	VARCHAR2(30)	Null		Default value for the USSGL Transaction Code Flexfield (for future use)
status_trx	VARCHAR2(30)	Null		The status of the transaction. If not populated, then defaulted from Transaction Type. Valid values are 'OP', 'CL', 'PEN', 'VD'.
paying_customer_id	NUMBER	Null		This column is required when the RECEIPT_METHOD_ID column is an automatic receipt method.
paying_site_use_id	NUMBER	Null		This column is required when the RECEIPT_METHOD_ID column is an automatic receipt method.
doc_sequence_value	NUMBER(15)	Null		Document Number. Must not exist in Oracle Receivables.
attribute_category	VARCHAR2(30)	Null		Descriptive flexfield structure definition column.
attribute1 - 10	VARCHAR2(150)	Null		Descriptive flexfield segment.
global_attribute_category	VARCHAR2(30)	Null		Reserved for country-specific functionality. (For future use.)

Attribute Name	Data Type	Required	Default Value	Description
global_attribute1-30	VARCHAR2(150)	Null		Reserved for country-specific functionality. (For future use.)
interface_header_context	VARCHAR2(30)	Null		Interface header context.
interface_header_attribute1 - 15	VARCHAR2(30)	Null		Interface header attribute value.

P_TRX_LINES_TBL Parameter

The P_TRX_LINES_TBL parameter is of PL/SQL table type TRX_LINE_REC_TYPE.

TRX_LINE_REC_TYPE has the following attributes, as described in this table:

Attribute Name	Data Type	Required	Default Value	Description
trx_header_id	NUMBER	Yes		Identifier for the Invoice header record. This column can be generated based on a sequence or any number value. The value does not get recorded into any table. This column ties back with P_TRX_HEADER_TBL.
trx_line_id	NUMBER	Yes		Identifier for the Invoice lines record. This column can be generated based on a sequence or any number value. The value does not get recorded into any table.
link_to_trx_line_id	NUMBER			This column is required only if line type is 'TAX' and 'FREIGHT' (if it is associated with any line).
line_number	NUMBER	Yes		Line number of the invoice
reason_code	VARCHAR2(30)			Reason code. Validated against AR_LOOKUPS.LOOKUP_TYPE = 'INVOICING_REASON'.

Attribute Name	Data Type	Required	Default Value	Description
inventory_item_id	NUMBER			Inventory item identifier. Mutually exclusive with the column MEMO_LINE_ID. Validated against mtl_system_items.inventory_item_id and invoice_enabled_flag = 'Y'.
description	VARCHAR2(240)			Line description. Required if inventory_item_id or memo_line_id is not provided.
quantity_ordered	NUMBER			Quantity of an order
quantity_invoiced	NUMBER			Quantity of invoice line. Required for Invoices.
unit_standard_price	NUMBER			List price per unit.
unit_selling_price	NUMBER			Selling price per unit for a transaction line. Required for Invoices.
sales_order	VARCHAR2(50)			Sales order number for this transaction.
sales_order_line	VARCHAR2(30)			Sales order line number for this transaction.
sales_order_date	DATE			Sales order date for this transaction.
accounting_rule_id	NUMBER			Accounting rule identifier. Must provide a value for invoice with Rule ID. Validated against RA_RULES.
line_type	VARCHAR2(20)	Yes		Receivables lookup code for STD_LINE_TYPE.
attribute_category	VARCHAR2(30)			Descriptive flexfield structure definition column.
attribute1-15	VARCHAR2(150)			Descriptive flexfield segment.
rule_start_date	DATE			First GL date of the invoice. Only used for invoice with rules.

Attribute Name	Data Type	Required	Default Value	Description
interface_line_context	VARCHAR2(30)			Interface line context.
interface_line_attribute1-15	VARCHAR2(30)			Interface line attribute value.
sales_order_source	VARCHAR2(50)			The source of the sales order.
amount	NUMBER			Transaction line revenue amount. If line type = 'FREIGHT' or 'TAX', then amount must be populated.
tax_precedence	NUMBER			This column is obsolete and should not be populated.
tax_rate	NUMBER			Tax rate for a line. Required for TAX line in case amount is not populated.
memo_line_id	NUMBER			Memo line description identifier. Mutually exclusive with the column INVENTORY_ITEM_ID. Not required for 'TAX' and 'FREIGHT' lines.
uom_code	VARCHAR2(30)			Unit of measure code. Required for line type of 'LINE' and has an item on the line. Not required for 'TAX' and 'FREIGHT' lines.
default_ussgl_transaction_code	VARCHAR2(30)			Default value for the USSGL Transaction Code Flexfield. (For future use.)
default_ussgl_trx_code_context	VARCHAR2(30)			Default context value for the USSGL Transaction Code Flexfield. (For future use.)
vat_tax_id	NUMBER			Unique identifier for AR_VAT_TAX. Required for 'TAX' Lines.
tax_exempt_flag	VARCHAR2(1)			Tax Lines are controlled by the lookup (TAX_CONTROL_FLAG), which allows for standard tax, exempt tax, and required tax.

Attribute Name	Data Type	Required	Default Value	Description
tax_exempt_number	VARCHAR2(80)			Exemption certificate number for item lines that have TAX_EXEMPT_FLAG set to E for exempt.
tax_exempt_reason_code	VARCHAR2(30)			Tax Exempt Reason, for item lines that have tax_exempt_flag set to "E" (exempt).
movement_id	NUMBER			Intrastate movement ID number that is tied to the shipment information.
global_attribute1-20-20	VARCHAR2(150)			Reserved for country-specific functionality. (For future use.)
global_attribute_category	VARCHAR2(30)			Reserved for country-specific functionality. (For future use.)
amount_includes_tax_flag	VARCHAR2(1)			Y indicates tax is inclusive. N indicates tax is exclusive. NULL for lines indicates tax cannot be overridden or tax is a tax group. Cannot be NULL for tax types. Must be NULL for other types.
warehouse_id	NUMBER			Foreign key to the HR_ORGANIZATIONS table. The warehouse identifies the ship-from location and can be used to control taxation. Within the US, the Warehouse ID is important when calculating tax on the origin/modified origin state sales tax. Outside the US you can use tax groups and conditions to build a schedule of multiple conditional taxes based on both the ship-from and ship-to county/county/state or provinces.
contract_line_id	NUMBER			Identifies the contract line from Oracle Contracts Core that is associated with this line.
source_data_key1-5	VARCHAR2(150)			Identifies source data from original system.
invoiced_line_acctg_level	VARCHAR2(15)			Identifies accounting level for invoiceable lines in original system.

P_TRX_DIST_TBL Parameter

The P_TRX_DIST_TBL parameter is of PL/SQL table type TRX_DIST_REC_TYPE.

TRX_DIST_REC_TYPE has the following attributes, as described in this table:

Attribute Name	Data Type	Required	Default Value	Description
trx_line_id	NUMBER	Yes		Identifier for the Invoice lines record. This column can be generated based on a sequence or any number value. The value does not get recorded into any table.
trx_header_id	NUMBER			Identifier for the Invoice header record. This column can be generated based on a sequence or any number value. The value does not get recorded into any table. This column ties back with P_TRX_HEADER_TBL. Required in case of 'REC' distribution type.
trx_dist_id	NUMBER	Yes		Identifier for the Distribution record. This column can be generated based on a sequence or any number value. The value does not get recorded into any table.
account_class	VARCHAR2(20)	Yes		Account Class for this distribution. Freight, Receivable, Revenue, AutoInvoice Clearing, Tax, Unbilled Receivable, Unearned Revenue, or Charges account type.
amount	NUMBER			Amount of this record in the foreign currency. Required if percentage is not passed.
acctd_amount	NUMBER			Amount of this record in the functional currency. If not populated, then it will be populated based on amount passed.
percent	NUMBER			Percent of the line amount represented by this record. Required if amount is not passed.
code_combination_id	NUMBER	Yes		Code combination ID for Accounting Flexfield. Validated against gl_code_combinations.code_combination_id.

Attribute Name	Data Type	Required	Default Value	Description
attribute_category	VARCHAR2(30)			Descriptive flexfield structure definition column.
attribute1-15	VARCHAR2(150)			Descriptive flexfield segment.
comments	VARCHAR2(240)			Comment about the revenue distribution.

P_TRX_SALESCREDITS_TBL Parameter

The P_TRX_SALESCREDITS_TBL parameter is of PL/SQL table type TRX_SALESCREDITS_REC_TYPE.

TRX_SALESCREDITS_REC_TYPE has the following attributes, as described in this table:

Attribute Name	Data Type	Required	Default Value	Description
trx_salescredit_id	NUMBER	Yes		Identifier for the Salesperson on the lines record. This column can be generated based on a sequence or any number value. The value does not get recorded into any table.
trx_line_id	NUMBER	Yes		Identifier for the Invoice lines record. This column can be generated based on a sequence or any number value. The value does not get recorded into any table.
salesrep_id	NUMBER	Yes		Identifies the salesperson for this sales credit assignment. Validated against ra_salesreps.salesrep_id.
salesrep_number	VARCHAR2(30)			Salesrep Number assignment. Validated against ra_salesreps.salesrep_number. If both number and ID is passed, then ID will take precedence.
sales_credit_type_name	VARCHAR2(30)			Sales Credit Type Name. Validated against so_sales_credit_types.name.

Attribute Name	Data Type	Required	Default Value	Description
sales_credit_type_id	NUMBER	Yes		Sales Credit Type Identifier. Validated against so_sales_credit_types.sales_credit_type_id. If both ID and name are passed, then ID will take precedence.
salescredit_amount_split	NUMBER			The amount of revenue/non-revenue credit for this salesperson/customer. Required if salescredit_percent_split is not passed.
salescredit_percent_s	NUMBER			The percent of revenue/non-revenue credit for this salesperson/customer. Required if salescredit_amount_split is not passed.
attribute_category	VARCHAR2(30)			Descriptive flexfield structure definition column.
attribute1-15	VARCHAR2(150)			Descriptive flexfield segment.

Example for Creating Multiple Invoices in a Batch

Objective:

To create an Invoice using a call to ar_invoice_api_pub.Create_invoive and passing a minimum number of Input parameters.

1. DECLARE

```

l_return_status          varchar2(1);
l_msg_count              number;
l_msg_data                varchar2(2000);
l_batch_id                number;

l_batch_source_rec        ar_invoice_api_pub.batch_source_rec_type;
l_trx_header_tbl          ar_invoice_api_pub.trx_header_tbl_type;
l_trx_lines_tbl            ar_invoice_api_pub.trx_line_tbl_type;
l_trx_dist_tbl             ar_invoice_api_pub.trx_dist_tbl_type;
l_trx_salescredits_tbl    ar_invoice_api_pub.trx_salescredits_tbl_type;

CURSOR cBatch IS
    select customer_trx_id
    from ra_customer_trx_all
    where batch_id = l_batch_id;

CURSOR cValidTxn IS
    SELECT trx_header_id
    From ar_trx_header_gt
    WHERE trx_header_id not in (
        SELECT trx_header_id
        FROM ar_trx_errors_gt);

```

2. BEGIN

1. Set applications context if not already set.

```
    fnd_global.apps_initialize(1318, 50559, 222,0);
```

2. Populate header information.

```

    l_trx_header_tbl(1).trx_header_id := 101;
    l_trx_header_tbl(1).trx_number := 'Test Invoice API';
    l_trx_header_tbl(1).bill_to_customer_id := 1006;
    l_trx_header_tbl(1).cust_trx_type_id := 2376;

```

3. Populate batch source information.

```
    l_batch_source_rec.batch_source_id := 1188;
```

4. Populate line 1 information.

```

    l_trx_lines_tbl(1).trx_header_id := 101;
    l_trx_lines_tbl(1).trx_line_id := 101;
    l_trx_lines_tbl(1).line_number := 1;
    l_trx_lines_tbl(1).memo_line_id := 8;
    l_trx_lines_tbl(1).quantity_invoiced := 10;
    l_trx_lines_tbl(1).unit_selling_price := 12;
    l_trx_lines_tbl(1).line_type := 'LINE';

```

5. Populate line 2 information.

```

    l_trx_lines_tbl(2).trx_header_id := 101;
    l_trx_lines_tbl(2).trx_line_id := 102;
    l_trx_lines_tbl(2).line_number := 2;
    l_trx_lines_tbl(2).description := 'Test';
    l_trx_lines_tbl(2).quantity_invoiced := 12;
    l_trx_lines_tbl(2).unit_selling_price := 12;
    l_trx_lines_tbl(2).line_type := 'LINE';

```

6. Populate freight information and link it to line 1.

```

l_trx_lines_tbl(3).trx_header_id := 101;
l_trx_lines_tbl(3).trx_line_id := 103;
l_trx_lines_tbl(3).link_to_trx_line_id := 101;
l_trx_lines_tbl(3).line_number := 1;
l_trx_lines_tbl(3).line_type := 'FREIGHT';
l_trx_lines_tbl(3).amount := 25;

```

7. Call the invoice api to create multiple invoices in a batch.

```

AR_INVOICE_API_PUB.create_invoice(
    p_api_version          => 1.0,
    p_batch_source_rec     => l_batch_source_rec,
    p_trx_header_tbl       => l_trx_header_tbl,
    p_trx_lines_tbl        => l_trx_lines_tbl,
    p_trx_dist_tbl         => l_trx_dist_tbl,
    p_trx_salescredits_tbl => l_trx_salescredits_tbl,
    x_return_status         => l_return_status,
    x_msg_count             => l_msg_count,
    x_msg_data              => l_msg_data);

IF l_return_status = fnd_api.g_ret_sts_error OR
    l_return_status = fnd_api.g_ret_sts_unexp_error THEN
    dbms_output.put_line('unexpected errors found!');
ELSE

```

8. Check if there are record exist in error table. If no records exist for a `trx_header_id`, then only Invoice will create in the system; otherwise not.

```

For cValidTxnRec IN cvalidTxn
loop
    IF (ar_invoice_api_pub.g_api_outputs.batch_id IS NOT NULL)
THEN
    dbms_output.put_line('Invoice(s) successfully created!') ;
    dbms_output.put_line('Batch ID: ' ||
ar_invoice_api_pub.g_api_outputs.batch_id);
    l_batch_id := ar_invoice_api_pub.g_api_outputs.batch_id;

```

9. To see all `customer_trx_id` for this batch:

```

for cBatchRec in cBatch
loop
    dbms_output.put_line ( 'Cust Trx Id ' ||
cBatchRec.customer_trx_id );
    end loop;
    ELSE
        dbms_output.put_line('Errors found!');
    END IF;
End loop;
END IF;
END;
/

```

10. See all the validation errors.

```

SET LINESIZE 200
COLUMN trx_header_id HEADING 'Header ID'
COLUMN trx_line_id HEADING 'Line ID'
COLUMN error_message HEADING 'Message'
COLUMN invalid_value HEADING 'Invalid Value'
COLUMN trx_header_id FORMAT 9999999
COLUMN trx_line_id FORMAT 9999999
COLUMN error_message FORMAT a30
COLUMN invalid_value FORMAT a20
SELECT trx_header_id, trx_line_id, error_message, invalid_value
FROM ar_trx_errors_gt;

```

Example for Creating a Single Invoice

Objective:

To create an Invoice using a call to ar_invoice_api_pub.Create_single_invoive and passing a minimum number of Input parameters.

1. DECLARE

```

l_return_status      varchar2(1);
l_msg_count         number;
l_msg_data          varchar2(2000);
l_batch_id          number;
l_cnt               number := 0;

l_batch_source_rec  ar_invoice_api_pub.batch_source_rec_type;
l_trx_header_tbl    ar_invoice_api_pub.trx_header_tbl_type;
l_trx_lines_tbl     ar_invoice_api_pub.trx_line_tbl_type;
l_trx_dist_tbl      ar_invoice_api_pub.trx_dist_tbl_type;
l_trx_salescredits_tbl ar_invoice_api_pub.
l_customer_trx_id   number;

```

2. BEGIN

1. Set applications context if not already set.

```
fnd_global.apps_initialize(1318, 50559, 222,0);
```

2. Populate header information.

```

l_trx_header_tbl(1).trx_header_id := 101;
l_trx_header_tbl(1).trx_number := 'Test Invoice API';
l_trx_header_tbl(1).bill_to_customer_id := 1006;
l_trx_header_tbl(1).cust_trx_type_id := 2376;

```

3. Populate batch source information.

```
l_batch_source_rec.batch_source_id := 1188;
```

4. Populate line 1 information.

```

l_trx_lines_tbl(1).trx_header_id := 101;
l_trx_lines_tbl(1).trx_line_id := 101;
l_trx_lines_tbl(1).line_number := 1;
l_trx_lines_tbl(1).memo_line_id := 8;
l_trx_lines_tbl(1).quantity_invoiced := 10;
l_trx_lines_tbl(1).unit_selling_price := 12;
l_trx_lines_tbl(1).line_type := 'LINE';

```

5. Populate line 2 information.

```

l_trx_lines_tbl(2).trx_header_id := 101;
l_trx_lines_tbl(2).trx_line_id := 102;
l_trx_lines_tbl(2).line_number := 2;
l_trx_lines_tbl(2).description := 'Test';
l_trx_lines_tbl(2).quantity_invoiced := 12;
l_trx_lines_tbl(2).unit_selling_price := 12;
l_trx_lines_tbl(2).line_type := 'LINE';

```

6. Populate freight information and link it to line 1.

```

l_trx_lines_tbl(3).trx_header_id := 101;
l_trx_lines_tbl(3).trx_line_id := 103;
l_trx_lines_tbl(3).link_to_trx_line_id := 101;
l_trx_lines_tbl(3).line_number := 1;
l_trx_lines_tbl(3).line_type := 'FREIGHT';
l_trx_lines_tbl(3).amount := 25;

```

7. Call the invoice api to create multiple invoices in a batch.

```

AR_INVOICE_API_PUB.create_single_invoice(
  p_api_version          => 1.0,
  p_batch_source_rec     => l_batch_source_rec,
  p_trx_header_tbl       => l_trx_header_tbl,
  p_trx_lines_tbl        => l_trx_lines_tbl,
  p_trx_dist_tbl         => l_trx_dist_tbl,
  p_trx_salescredits_tbl => l_trx_salescredits_tbl,
  x_customer_trx_id      => l_customer_trx_id,
  x_return_status         => l_return_status,
  x_msg_count             => l_msg_count,
  x_msg_data              => l_msg_data);

IF l_return_status = fnd_api.g_ret_sts_error OR
   l_return_status = fnd_api.g_ret_sts_unexp_error THEN
  dbms_output.put_line('unexpected errors found!');
ELSE

```

8. Check whether any record exist in error table

```

SELECT count(*)
  Into      cnt
  From ar_trx_errors_gt;
  IF cnt = 0
  THEN
    dbms_output.put_line ( 'Customer Trx id '|| l_customer_trx_id);

  ELSE
    dbms_output.put_line ( 'Transaction not Created, Please check
ar_trx_errors_gt table');
    END IF;

  END;
/

```

9. See all the validation errors.

```

SET LINESIZE 200
COLUMN trx_header_id HEADING 'Header ID'
COLUMN trx_line_id    HEADING 'Line ID'
COLUMN error_message  HEADING 'Message'
COLUMN invalid_value  HEADING 'Invalid Value'
COLUMN trx_header_id FORMAT 9999999
COLUMN trx_line_id    FORMAT 9999999
COLUMN error_message  FORMAT a30
COLUMN invalid_value  FORMAT a20
SELECT trx_header_id, trx_line_id, error_message, invalid_value
FROM ar_trx_errors_gt;

```

Note: In the above examples, we did not pass distribution, sales credits, or contingencies . Note, however, that you *can* create an invoice passing distributions, sales credits, and contingencies.

Prepayments API User Notes

Overview

This document outlines the specifications and the methodology for using the Prepayments API.

Use the Prepayments API to:

- Generate a unique payment grouping identifier (payment_set_id)
- Create a prepayment receipt flagged with this payment_set_id
- Apply the prepayment receipt to a receivable activity of type Prepayment

You can access this API:

- As standard PL/SQL server-side routine calls
- Through forms, utilizing the capability of Forms6 to have a procedure as its underlying base table

Basic Business Needs

The Prepayments API addresses the following business needs:

- Enables the creation of a receipt in advance of the invoicing event
- Provides a mechanism of matching a prepayment receipt to a prepaid invoice

The Prepayments API lets you model down payments, deposits, or prepayments as receipts created in Oracle Receivables in advance of the invoice creation event.

It is not intended for the purpose of creating receipts for existing invoices, simply before the invoices.

API Usage

This section describes how to use the Prepayments API to:

- Create a prepayment receipt
- Apply the prepayment receipt to the prepayment activity
- Calculate the amount of all the installments of a particular payment term

To create, apply, and refund a prepayment receipt, you can call the following PL/SQL routine:

- **AR_PREPAYMENTS_PUB.Create_Prepayment**, page 7-2: Use this routine to create a prepayment receipt.
- **AR_PREPAYMENTS_PUB.Get_Installment**, page 7-9: Use this routine to calculate the amount of all installments of a given payment term.

AR_PREPAYMENTS_PUB.Create_Prepayment

This routine is called to create a prepayment receipt.

This API routine has 5 output, 8 input-output, and 56 input parameters. Of the output parameters, the API returns 5.

Input

Standard API parameters: 4

Prepayment parameters: 48 + 8 (INOUT) parameters

4 (global descriptive flexfield parameters)

Output

Standard API parameters: 3

Prepayment parameters: 2 + 8 (INOUT) parameters

Parameter Descriptions

The input descriptive flexfield parameter is a record of type *attribute_rec_type*.

```

TYPE attribute_rec_type IS RECORD(
    attribute_category      VARCHAR2(30) DEFAULT NULL,
    attribute1              VARCHAR2(150) DEFAULT NULL,
    attribute2              VARCHAR2(150) DEFAULT NULL,
    attribute3              VARCHAR2(150) DEFAULT NULL,
    attribute4              VARCHAR2(150) DEFAULT NULL,
    attribute5              VARCHAR2(150) DEFAULT NULL,
    attribute6              VARCHAR2(150) DEFAULT NULL,
    attribute7              VARCHAR2(150) DEFAULT NULL,
    attribute8              VARCHAR2(150) DEFAULT NULL,
    attribute9              VARCHAR2(150) DEFAULT NULL,
    attribute10             VARCHAR2(150) DEFAULT NULL,
    attribute11             VARCHAR2(150) DEFAULT NULL,
    attribute12             VARCHAR2(150) DEFAULT NULL,
    attribute13             VARCHAR2(150) DEFAULT NULL,
    attribute14             VARCHAR2(150) DEFAULT NULL,
    attribute15             VARCHAR2(150) DEFAULT NULL);

```

The input global descriptive flexfield parameter is a record of type *global_attr_rec_type*.

```

TYPE global_attribute_rec_type IS RECORD(
    global_attribute_category      VARCHAR2(30) default null,
    global_attribute1              VARCHAR2(150) default NULL,
    global_attribute2              VARCHAR2(150) DEFAULT NULL,
    global_attribute3              VARCHAR2(150) DEFAULT NULL,
    global_attribute4              VARCHAR2(150) DEFAULT NULL,
    global_attribute5              VARCHAR2(150) DEFAULT NULL,
    global_attribute6              VARCHAR2(150) DEFAULT NULL,
    global_attribute7              VARCHAR2(150) DEFAULT NULL,
    global_attribute8              VARCHAR2(150) DEFAULT NULL,
    global_attribute9              VARCHAR2(150) DEFAULT NULL,
    global_attribute10             VARCHAR2(150) DEFAULT NULL,
    global_attribute11             VARCHAR2(150) DEFAULT NULL,
    global_attribute12             VARCHAR2(150) DEFAULT NULL,
    global_attribute13             VARCHAR2(150) DEFAULT NULL,
    global_attribute14             VARCHAR2(150) DEFAULT NULL,
    global_attribute15             VARCHAR2(150) DEFAULT NULL,
    global_attribute16             VARCHAR2(150) DEFAULT NULL,
    global_attribute17             VARCHAR2(150) DEFAULT NULL,
    global_attribute18             VARCHAR2(150) DEFAULT NULL,
    global_attribute19             VARCHAR2(150) DEFAULT NULL,
    global_attribute20             VARCHAR2(150) DEFAULT NULL);

```

The following table lists the parameters that pertain specifically to the Create Prepayment routine:

Parameter	Type	Mandatory/Optional	Data-type	Default Value	Description
p_api_version	IN	M	NUMBER	Constant 1.0	
p_init_msg_list	IN	O	VARCHAR2	Default FND_API.G_FALSE	
p_commit	IN	O	VARCHAR2	Default FND_API.G_FALSE	

Parameter	Type	Mandatory/Optional	Data-type	Default Value	Description
p_validation_level	IN	O	NUMBER	Default FND_API.G_VALID_LEVEL_FULL	
x_return_status	OUT	M	VARCHAR2		Return status of the prepayment call
x_msg_count	OUT	M	NUMBER		Message counts in message stack
x_msg_data	OUT	M	VARCHAR2		Message text in message stack.
p_usr_currency_code	IN	O	VARCHAR2		Translated currency code
p_currency_code	IN	M	VARCHAR2		Currency of the receipt
p_usr_exchange_rate_type	IN	O	VARCHAR2		User exchange rate type
p_exchange_rate_type	IN	O	VARCHAR2		Exchange rate type, if other than functional currency (if functional currency is different than receipt)
p_exchange_rate_date	IN	O	DATE		Exchange rate date
p_exchange_rate	IN	O	NUMBER		Exchange rate
p_amount	IN	M	NUMBER		Receipt amount
p_factor_discount_amount	IN	O	NUMBER		Factor discount amount
p_receipt_number	IN OUT	O	VARCHAR2		Receipt number, need to pass if doc sequence is not enabled
p_receipt_date	IN	O	DATE		Receipt creation Date
p_gl_date	IN	O	DATE		GL date of the receipt
p_maturity_date	IN	O	DATE		Maturity date of the receipt

Parameter	Type	Mandatory/Optional	Data-type	Default Value	Description
p_postmark_date	IN	O	DATE		Postmark date of receipt
p_customer_id	IN	M	NUMBER		Customer ID of the receipt
p_customer_name	IN	O	VARCHAR2		Customer Name
p_customer_number	IN	O	NUMBER		Customer Number
p_customer_bank_account_id	IN	M	NUMBER		Customer bank account ID
p_customer_bank_account_number	IN	O	VARCHAR2		Customer bank account number
p_customer_bank_account_name	IN	O	VARCHAR2		Customer bank account name
p_location	IN	O	VARCHAR2		Location
p_customer_site_use_id	IN	M	NUMBER		Site use ID
p_customer_receipt_reference	IN	O	VARCHAR2		Reference information on receipt header
p_override_remit_account_flag	IN	O	VARCHAR2		Remittance account override flag
p_remittance_bank_account_id	IN	M	VARCHAR2		Remittance bank account ID
p_remittance_bank_account_number	IN	O	VARCHAR2		Remittance bank account number
p_remittance_bank_account_name	IN	O	VARCHAR2		Remittance bank account name

Parameter	Type	Mandatory/Optional	Data-type	Default Value	Description
p_deposit_date	IN	O	DATE		Deposit date
p_receipt_method_id	IN	M	NUMBER		Remittance method ID (receipt method)
p_receipt_method_name	IN	O	VARCHAR2		Receipt method name
p_doc_sequence_value	IN	O	NUMBER		Doc sequence value, if doc sequence is enabled (mandatory if doc sequence is enabled)
p_ussgl_transaction_code	IN	O	NUMBER		USSGL transaction code, if exists, on receipt header
p_anticipated_clearing_date	IN	O	DATE		Anticipated receipt clearing date
p_called_from	IN	M	NUMBER		Which program called this routine?
p_attribute_rec	IN	O	RECORD TYPE		Receipt Header attributes
p_global_attributes_rec	IN	O	RECORD TYPE		Global attributes on receipt header (GDF)
p_receipt_comments	IN	O	VARCHAR2		Receipt header comments
p_issuer_name	IN	O	VARCHAR2		AR Notes Issuer name
p_issue_date	IN	O	DATE		AR Notes Issue Date
p_issuer_bank_branch_id	IN	O	NUMBER		AR Notes Issuer bank branch ID
p_cr_id	OUT	M	NUMBER		Cash receipt ID
p_applied_payment_schedule_id	IN	M	NUMBER		For prepayment, it will be -7

Parameter	Type	Mandatory/Optional	Data-type	Default Value	Description
p_amount_applied	IN	O	NUMBER		Specify amount which needs to be put in prepayment out of the receipt amount
p_application_ref_type	IN	O	VARCHAR2		Prepayment application reference from a lookup code for lookup type AR_PREPAYMENT_TYPE to indicate where it is created from. For example, OM.
p_application_ref_id	IN OUT	M	NUMBER		Application reference ID. For example, order ID.
p_application_ref_num	IN OUT	M	VARCHAR2		Reference number. For example, order number.
p_secondary_application_ref_id	IN OUT	O	NUMBER		Additional reference, if exists
p_receivable_trx_id	IN	O	NUMBER		Receivable activity ID, default if not passed for prepayment.
p_amount_applied_from	IN	O	NUMBER		Amount applied in functional currency
p_apply_date	IN	O	DATE		If null, takes sysdate
p_apply_gl_date	IN	O	DATE		Application GL date
app_ussql_transaction_code	IN	O	VARCHAR2		USSQL transaction type code on application
p_show_closed_invoices	IN	O	VARCHAR2		Default FALSE
p_move_deferred_tax	IN	O	VARCHAR2		Default Y
app_attribute_rec	IN	O	RECORD TYPE		Application attributes

Parameter	Type	Mandatory/Optional	Data-type	Default Value	Description
app_global_attribute_rec	IN	O	RECORD TYPE		Global application attributes (GDF)
app_comments	IN	O	VARCHAR2		comments on application
p_payment_server_order_num	IN OUT	M	VARCHAR2		Payment server order number
p_call_payment_processor	IN	O	VARCHAR2		Decides whether to call Oracle Payments. DEFAULT FND_API.G_FALSE
p_payment_response_error_code	IN OUT	M	VARCHAR2		Oracle Payments return error code
p_approval_code	IN OUT	M	VARCHAR2		Credit Card Approval code
p_receivable_application_id	OUT	M	NUMBER		Receivable applications ID of the application
p_payment_set_id	IN OUT	M	NUMBER		If passed, it will take the passed payment_set_id while creating prepayment application. Otherwise, generate a new number and pass it back.

Example

The following is a test case for creating a prepayment.

Objective:

To create a prepayment, passing the minimum number of parameters.

Entered parameters:

- p_api_version
- p_currency_code
- p_amount
- p_customer_id

- p_customer_bank_account_id
- p_customer_site_use_id
- p_remittance_bank_account_id
- p_receipt_method_id
- p_called_from
- p_applied_payment_schedule_id
- p_application_ref_id
- p_application_ref_num

The API call in this case would be:

```
AR_PREPAYMENTS_PUB.create_prepayment(
    p_api_version          => 1.0,
    p_commit                => FND_API.G_FALSE,
    x_return_status          => x_return_status,
    x_msg_count              => x_msg_count,
    x_msg_data                => x_msg_data,
    p_init_msg_list          => FND_API.G_TRUE,
    p_receipt_number          => l_receipt_number,
    p_currency_code          => l_currency_code,
    p_amount                 => p_payment_amount,
    p_receipt_method_id      => l_receipt_method_id,
    p_customer_id              => p_customer_id,
    p_customer_site_use_id    => l_site_use_id,
    p_customer_bank_account_id => p_bank_account_id,
    p_currency_code          => l_receipt_currency_code,
    p_exchange_rate           => l_receipt_exchange_rate,
    p_exchange_rate_type      => l_receipt_exchange_rate_type,
    p_exchange_rate_date      => l_receipt_exchange_rate_date,
    p_applied_payment_schedule_id => p_payment_schedule_id,
    p_application_ref_type    => l_application_ref_type, --Order type
    p_application_ref_num     => l_application_ref_num, --Order Number
    p_application_ref_id      => l_application_ref_id, --Order Id
    p_cr_id                  => l_cr_id --OUT,
    p_receivable_application_id => l_receivable_application_id --OUT
    p_call_payment_processor  => l_call_payment_processor
    p_payment_response_error_code => l_payment_response_error_code
    p_payment_set_id          => l_payment_set_id -If not passed generate a new
    number
);
```

AR_PREPAYMENTS_PUB.Get_Installment

This routine is called to calculate the amount of all installments of a given payment term.

This API routine has 4 output and 5 input parameters. Of the output parameters, the API returns 5.

Input

Standard API parameters: 0

Prepayment parameters: 5

Output

Standard API parameters: 3

Prepayment parameters: 1

Parameter Descriptions

The following table lists the parameters that pertain specifically to the Get Installment routine:

Parameter	Type	Mandatory/Optional	Data-Type	Default Value	Details
p_term_id	IN	M	NUMBER		Payment term ID
p_amount	IN	M	NUMBER		Line amount and additional charges (if any)
p_tax	IN	O	NUMBER		Tax amount
p_freight	IN	O	NUMBER		Freight charges
p_currency_code	IN	M	VARCHAR2		Currency code for calculating the installment amount
p_installment_tb	OUT	O	NUMBER		A table consisting of installment number and installment amount
x_return_status	OUT	M	VARCHAR2		Return status of the API call
x_msg_count	OUT	M	NUMBER		Message counts in message stack
x_msg_data	OUT	M	VARCHAR2		Message text in message stack.

Example

The following is a test case for get_installment.

Objective:

To get the installment amount given an amount, payment term and currency code.

Entered parameters:

- p_term_id
- p_amount
- p_currency_code

```
AR_PREPAYMENTS_PUB.get_installment(
    p_term_id      => l_term_id      ,
    p_amount        => l_amount,
    p_currency_code => l_currency_code,
    p_installment_tbl => l_installment_tbl ,  --OUT
    x_return_status  => x_return_status,
    x_msg_count      => x_msg_count,
    x_msg_data        => x_msg_data);
```

Messages

Messages play an important role in the effectiveness of your API calls. The right message is raised at the right point to convey to you the exact error that has occurred or any warnings that have been raised.

In the Prepayments API, all error messages and warnings raised during the execution are put on the message stack and can be retrieved by the user as described in Robust Validation, page 1-2.

The following is the list of all error messages raised by the Prepayments API.

Message Number	Message Name	Message Description
96735	AR_RAPI_CUS_BK_AC_2_INVALID	Invalid combination of customer bank account name and number.
294347	AR_RAPI_PREPAY_SEQ_FAILED	The prepayment sequence generation has failed. Please contact your system administrator.
	AR_PPAY_PAY_TERM_INVALID	Payment term ID is invalid.
	AR_PPAY_BASE_AMOUNT_INVALID	The amount can not be null, 0, or negative.
96734	AR_RAPI_CURR_CODE_INVALID	Currency code is invalid.

Since this API also calls the Receipt API AR_RECEIPT_API_PUB, it could also throw messages raised by the Receipt API.

Please refer to messages listed in Receipt API Messages, page 8-137.

Receipt API User Notes

Overview

This document outlines the specifications and the methodology for using the various Receipt APIs. These APIs provide an extension to existing functionality of creating and manipulating receipts through standard AR Receipts forms and lockboxes.

You can access these APIs:

- As standard PL/SQL server-side routine calls
- Through forms, utilizing the capability of Forms6 to have a procedure as its underlying base table

Basic Business Needs

The Receipt API provides the following basic functionality via different API calls:

- Creating a cash receipt
- Applying a cash receipt to a debit item
- Creating a cash receipt and applying it to a debit item in one pass
- On-account application
- Unapplying the on-account application
- Unapplying the receipt application to a particular transaction
- Reversing the receipt
- Activity application, such as Receipt Write-off

- Creating a miscellaneous receipt
- Other account application, such as Claim Investigation
- Receipt-to-receipt application
- Creating a cash receipt and an on-account application in one pass

Integration with Oracle Payments

The following table illustrates the integration between Oracle Payments and the Receipt API routines that create receipts:

Receipt API Routine	Calls Oracle Payments?
Ar_receipt_api_pub.Create_cash	No
Ar_receipt_api_pub.Create_and_apply	Yes
Ar_receipt_api_pub.Create_misc	No
Ar_receipt_api_pub.Create_apply_on_acc	Yes

API Usage

To create, apply, unapply, or reverse a cash receipt, you can call the following PL/SQL APIs:

- Ar_receipt_api_pub.Create_cash, page 8-3: Creates a single cash receipt, as in the case of manually created cash receipts.
- Ar_receipt_api_pub.Apply, page 8-20: Applies a cash receipt to a particular installment of a debit item. The application can also be a cross currency application.
- Ar_receipt_api_pub.Create_and_apply, page 8-34: Creates a cash receipt and applies it to a specified installment of a debit item in one pass. Application fails if the creation fails due to some reason.
- Ar_receipt_api_pub.Unapply, page 8-54: Unapplies the application of a particular installment of a debit item against the specified cash receipt.
- Ar_receipt_api_pub.Apply_on_account, page 8-60: Creates an on-account application for a cash receipt.

- `Ar_receipt_api_pub.Unapply_on_account`, page 8-65: Unapplies the on-account application on the specified receipt.
- `Ar_receipt_api_pub.Reverse`, page 8-69: Reverses the specified receipt.
- `Ar_receipt_api_pub.activity_application`, page 8-75: Applies to an activity, such as Receipt Write-off.
- `Ar_receipt_api_pub.activity_unapplication`, page 8-83: Unapplies from an activity, such as a Receipt Write-off.
- `Ar_receipt_api_pub.Create_misc`, page 8-87: Creates a single miscellaneous receipt.
- `Ar_receipt_api_pub.apply_other_account`, page 8-101: Applies to other account activities, such as Claim Investigation (for Trade Management customers only).
- `Ar_receipt_api_pub.unapply_other_account`, page 8-108: Unapplies from other account activities, such as Claim Investigation.
- `Ar_receipt_api_pub.apply_open_receipt`, page 8-112: Creates a receipt-to-receipt application (payment netting).
- `Ar_receipt_api_pub.unapply_open_receipt`, page 8-119: Unapplies a receipt-to-receipt application.
- `Ar_receipt_api_pub.Create_apply_on_acc`, page 8-121: Creates a cash receipt and an on-account application in one pass. If the receipt creation fails, then the application fails as well.

Ar_receipt_api_pub.Create_cash

This routine is called to create cash receipts for the payment received in the form of a check or cash. Cash receipts can be created as identified (with a customer) or as unidentified (without a customer).

Note: This routine does *not* call Oracle Payments directly. See Integration with Oracle Payments, page 8-2.

This API routine has 4 output and 47 input parameters in total. As one of the output parameters, the API returns the `cash_receipt_id` of the cash receipt created. The following is the breakdown of the parameters:

Input

Standard API parameters: 4

Cash Receipt parameters: 41 + 1 (descriptive flexfield parameter)

+ 1 (global descriptive flexfield parameter)

Output

Standard API parameters: 3

Cash Receipt parameters: 1

Parameter Descriptions

The input descriptive flexfield parameter is a record of type *attribute_rec_type*.

```
TYPE attribute_rec_type IS RECORD
  (p_attribute_category      IN VARCHAR2,
   p_attribute1              IN VARCHAR2,
   p_attribute2              IN VARCHAR2,
   p_attribute3              IN VARCHAR2,
   p_attribute4              IN VARCHAR2,
   p_attribute5              IN VARCHAR2,
   p_attribute6              IN VARCHAR2,
   p_attribute7              IN VARCHAR2,
   p_attribute8              IN VARCHAR2,
   p_attribute9              IN VARCHAR2,
   p_attribute10             IN VARCHAR2,
   p_attribute11             IN VARCHAR2,
   p_attribute12             IN VARCHAR2,
   p_attribute13             IN VARCHAR2,
   p_attribute14             IN VARCHAR2,
   p_attribute15             IN VARCHAR2);
```

The input global descriptive flexfield parameter is a record of type *global_attribute_rec_type*.

```
TYPE global_attribute_rec_type IS RECORD
  (p_global_attribute_category      IN VARCHAR2,
   p_global_attribute1              IN VARCHAR2,
   p_global_attribute2              IN VARCHAR2,
   p_global_attribute3              IN VARCHAR2,
   p_global_attribute4              IN VARCHAR2,
   p_global_attribute5              IN VARCHAR2,
   p_global_attribute6              IN VARCHAR2,
   p_global_attribute7              IN VARCHAR2,
   p_global_attribute8              IN VARCHAR2,
   p_global_attribute9              IN VARCHAR2,
   p_global_attribute10             IN VARCHAR2,
   p_global_attribute11             IN VARCHAR2,
   p_global_attribute12             IN VARCHAR2,
   p_global_attribute13             IN VARCHAR2,
   p_global_attribute14             IN VARCHAR2,
   p_global_attribute15             IN VARCHAR2,
   p_global_attribute16             IN VARCHAR2,
   p_global_attribute17             IN VARCHAR2,
   p_global_attribute18             IN VARCHAR2,
   p_global_attribute19             IN VARCHAR2,
   p_global_attribute20             IN VARCHAR2);
```

The following table lists standard API parameters that are common to all the routines in the Receipt API.

Parameter	Type	Data-type	Required	Default Value	Description
p_api_version	IN	NUMBER	Yes		Used to compare version numbers of incoming calls to its current version number. Unexpected error is raised if version incompatibility exists. In the current version of the API, you should pass in a value of 1.0 for this parameter.
p_init_msg_list	IN	VARCHAR 2		FND_API.G_FALSE	Allows API callers to request that the API does initialization of the message list on their behalf.
p_commit	IN	VARCHAR 2		FND_API.G_FALSE	Used by API callers to ask the API to commit on their behalf.
p_validation_level	IN	NUMBER		FND_API.G_VALID_LEVEL_FULL	Not to be used currently as this is a public API.
x_return_status	OUT	VARCHAR 2			Represents the API overall return status. Detailed in Return Status, page 1-4.
x_msg_count	OUT	NUMBER			Number of messages in the API message list
x_msg_data	OUT	VARCHAR 2			This is the message in encoded format if x_msg_count=1

The following table lists the parameters that pertain specifically to the cash receipt routine.

Note: If required parameters are not passed in a call to this API, then the call will fail. However, depending on the business scenario, you will have to pass in values for other parameters to successfully create the business object. Otherwise, error messages will be reported.

Parameter	Type	Data-type	Required	Description
p_usr_currency_code	IN	VARCHAR2		<p>The translated currency code.</p> <p>Used to derive the p_currency_code if it is not entered.</p> <p>Default: None</p> <p>Validation: Should be a valid currency, so that the corresponding currency code can be derived.</p> <p>Error: AR_RAPI_USR_CURR_CODE_INVALID</p>
p_currency_code	IN	VARCHAR2		<p>The actual currency code that gets stored in AR tables.</p> <p>Default:</p> <ol style="list-style-type: none"> 1. Derived from p_usr_currency_code if entered, else 2. Defaults to the functional currency code <p>Validation: Validated against the currencies in fnd_currencies table.</p> <p>Error: AR_RAPI_CURR_CODE_INVALID</p> <p>Warning: AR_RAPI_FUNC_CURR_DEFAULTED</p>
p_usr_exchange_rate_type	IN	VARCHAR2		<p>The translated exchange rate type.</p> <p>Used to derive the p_exchange_rate_type if it has not been entered.</p> <p>Default: None</p> <p>Validation: Should be a valid rate type.</p> <p>Error: AR_RAPI_USR_X_RATE_TYP_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_exchange_rate_ty pe	IN	VARCHAR2		<p>Exchange rate type stored in AR tables.</p> <p>Default:</p> <ol style="list-style-type: none"> 1. In case of foreign currency receipt, derived from p_usr_exchange_rate_type. 2. In case of foreign currency receipt, defaults from profile option 'AR: Default Exchange Rate Type' <p>Validation: Validated against values in gl_daily_conversion_types table.</p> <p>Error: AR_RAPI_X_RATE_TYPE_INVALID</p>
p_exchange_rate	IN	NUMBER		<p>The exchange rate between the receipt currency and the functional currency.</p> <p>Default:</p> <ol style="list-style-type: none"> 1. Derived from the Daily Rates table for rate_type <> 'User' in case of non-functional currency 2. If profile option Journals: Display Inverse Rate = 'Y', set user entered value to 1/ p_exchange_rate 3. The entered value is rounded to a precision of 38. <p>Validation:</p> <ol style="list-style-type: none"> 1. In case of non-functional currency the rate should have a positive value for rate type= 'User' 2. For non-functional currency and type is <> 'User', do not specify any value. <p>Error: AR_RAPI_X_RATE_INVALID AR_RAPI_X_RATE_NULL</p>

Parameter	Type	Data-type	Required	Description
p_exchange_rate_date	IN	DATE		<p>The date on which the exchange rate is valid.</p> <p>Default: Receipt date</p> <p>Validation: For a non-functional currency and type is \neq 'User' there should be a valid rate existing in the database for this date. This is a cross validation of type, currency, and date.</p> <p>Error: AR_NO_RATE_DATA_FOUND</p>
p_amount	IN	NUMBER	Yes	<p>The cash receipt amount.</p> <p>Default: Null</p> <p>Validation: >0</p> <p>Error: AR_RAPI_REC_AMT_NEGATIVE AR_RAPI_RCPT_AMOUNT_NULL</p>
p_factor_discount_amount	IN	NUMBER		<p>The bank charges on the cash receipt.</p> <p>Default: None</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. Bank charges not allowed if profile option AR: Create Bank Charges = 'No'. 2. Bank charges not allowed if the receipt state, derived from the receipt class of the receipt method \neq 'Cleared'. 3. If allowed then ≥ 0 <p>Error: AR_JG_BC_AMOUNT_NEGATIVE AR_BK_CH_NOT_ALLWD_IF_NOT_CLR</p>
p_receipt_number	IN	VARCHAR2(30)		<p>The receipt number of the receipt to be created.</p> <p>Default: If not specified, the receipt number is defaulted from the document sequence value.</p> <p>Validation: Receipt number should not be null.</p> <p>Error: AR_RAPI_RCPT_NUM_NULL</p>

Parameter	Type	Data-type	Required	Description
p_receipt_date	IN	DATE		<p>The receipt date of the entered cash receipt.</p> <p>Default: System date</p> <p>Validation: None</p> <p>Error: None</p>
p_gl_date	IN	DATE		<p>Date that this receipt will be posted to the General Ledger.</p> <p>Default: Gets defaulted to the receipt date if it is a valid gl_date.</p> <p>Validation: The date is valid if the following conditions are true:</p> <ul style="list-style-type: none"> • The date is in an Open or Future period • The period cannot be an Adjustment period <p>If the date is invalid, then:</p> <ul style="list-style-type: none"> • If the most recent open period is prior to the receipt date: last date of that period • If there is a period open after the receipt date: first date of the last open period <p>Error: AR_INVALID_APP_GL_DATE</p>
p_maturity_date	IN	DATE		<p>Receipt maturity date.</p> <p>Default: Deposit date</p> <p>Validation: \geq p_receipt_date</p> <p>Error: AR_RW_MAT_BEFORE_RCT_DATE</p>
p_postmark_date	IN	DATE		<p>The postmark date</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>

Parameter	Type	Data-type	Required	Description
p_customer_id	IN	NUMBER(15)		<p>The customer_id for the paying customer.</p> <p>Default: Defaulted from customer name/number</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. Customer exists and has prospect code = 'CUSTOMER' 2. Customer has a profile defined at the customer level <p>Error: AR_RAPI_CUST_ID_INVALID</p>
p_customer_name	IN	VARCHAR2(50)		<p>The name for the entered customer. Used to default the customer id if not specified.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPI_CUS_NAME_INVALID</p>
p_customer_number	IN			<p>The customer number. Used to default the customer_id if not specified</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPI_CUS_NUM_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_customer_bank_ac count_id	IN	NUMBER(15)		<p>The customer bank account id. Default: From bank account id/number</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. It must be a valid Bank Account of the paying customer 2. The inactive date (if defined) of the Bank Account, should be greater than the receipt_date 3. The receipt date has to be within the Start date and the End date of the Bank Account <p>Error: AR_RAPI_CUS_BK_AC_2_INVALID AR_RAPI_CUS_BK_AC_ID_INVALID</p>
p_customer_bank_ac count_num	IN	VARCHAR2(3 0)		<p>The customer bank account number. Used to default the customer bank account id, if not specified</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_customer_bank_ac count_name	IN	VARCHAR2(8 0)		<p>The customer bank account name. Used to default the customer bank account id, if not specified</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_location	IN	VARCHAR2(4 0)		<p>The Bill_To location for the customer. Used to derive the p_customer_site_use_id</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPI_CUS_LOC_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_customer_site_use_id	IN	NUMBER(15)		<p>The Bill_To site_use_id for the customer</p> <p>Default:</p> <ol style="list-style-type: none"> 1. Defaulted from customer location, else 2. Primary Bill_To customer site_use_id of the customer. <p>Validation: It should be a valid Bill_To site of the paying customer.</p> <p>Error: AR_RAPI_CUS_SITE_USE_ID_INVALID</p>
p_customer_receipt_reference	IN	VARCHAR2(30)		<p>This column is used to store a customer receipt reference value supplied by the customer at the confirmation time.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_override_remit_bank_account_flag	IN	VARCHAR2(1)		<p>The flag value decides when the remittance bank account can be overridden by the remittance selection process.</p> <p>Default: 'Y'</p> <p>Validation: valid values 'Y' and 'N'</p> <p>Error: AR_RAPI_INVALID_OR_REMIT_BK_AC</p>

Parameter	Type	Data-type	Required	Description
p_remittance_bank_account_id	IN	NUMBER(15)		<p>Identifies the user's bank account for depositing the receipt.</p> <p>Default:</p> <ol style="list-style-type: none"> From remittance bank account number From the receipt method based on logic mentioned in Defaulting, page 8-18. <p>Validation: Validation logic detailed in Validation, page 8-17.</p> <p>Error: AR_RAPI_Rem_BK_AC_ID_INVALID AR_RAPI_Rem_BK_AC_ID_NULL</p>
p_remittance_bank_account_num	IN	VARCHAR2(30)		<p>The remittance bank account number. Used to default the remittance bank account id if not specified.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPI_Rem_BK_AC_NUM_INVALID</p>
p_remittance_bank_account_name	IN	VARCHAR2(50)		<p>The remittance bank account name. Used to default the remittance bank account id if not specified</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPI_Rem_BK_AC_NAME_INVALID</p>
p_deposit_date	IN	DATE		<p>The deposit date.</p> <p>Default: receipt date</p> <p>Validation: None</p> <p>Error: None</p>

Parameter	Type	Data-type	Required	Description
p_receipt_method_id	IN	NUMBER(15)		<p>Identifies the receipt method of the receipt</p> <p>Default: From receipt method name</p> <p>Validation: Validation detailed in Validation, page 8-17</p> <p>Error: AR_RAPI_INVALID_RCT_MD_ID</p>
p_receipt_method_name	IN	VARCHAR2(30)		<p>The receipt method name of the receipt. Used to default the receipt method id if not specified.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPI_RCPT_MD_NAME_INVALID</p>
p_doc_sequence_value	IN	NUMBER		<p>Value assigned to document receipt.</p> <p>Default: Detailed in Defaulting, page 8-18</p> <p>Validation:</p> <ul style="list-style-type: none"> User should not pass in the value if the current document sequence is automatic. Document sequence value should not be entered if profile option Sequential Numbering is set to Not Used <p>Error: AR_RAPI_DOC_SEQ_AUTOMATIC AR_RAPI_DOC_SEQ_VAL_INVALID</p>
p_ussgl_transaction_code	IN	VARCHAR2(30)		<p>Code defined by public sector accounting.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_anticipated_clearing_date	IN	DATE		<p>Date the receipt is expected to be cleared.</p> <p>Default: None</p> <p>Validation: \geq gl_date</p> <p>Error: AR_RW_EFFECTIVE_BEFORE_GL_DATE</p>

Parameter	Type	Data-type	Required	Description
p_event	IN	VARCHAR2		<p>The event that resulted in the creation of the receipt. Currently used only by Bills Receivable.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_called_from	IN	VARCHAR2(20)		<p>This parameter is used to identify the calling routine. Currently used to identify only the 'BR_REMIT' program.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_attribute_record	IN	attribute_rec_type (PL/SQL defined record type)		<p>This is a record type which contains all the 15 descriptive flexfield segments and one descriptive flexfield structure defining column. It represents the Receipt Information flexfield.</p> <p>Default: DFF APIs used to do the defaulting and validation</p> <p>Validation: DFF APIs used to do the defaulting and validation</p> <p>Error: AR_RAPI_DESC_FLEX_INVALID</p>
p_global_attribute_record	IN	global_attribute_rec_type		<p>This is a record type which contains all the 20 global descriptive flexfield segments and one global descriptive flexfield structure defining column.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error:</p>
p_comments	IN	VARCHAR2(240)		User's comments

Parameter	Type	Data-type	Required	Description
p_issuer_name	IN	VARCHAR2(50)		<p>Issuer name of Notes Receivable (Asia Pacific Requirement)</p> <p>Default: None</p> <p>Validation: None</p> <p>Error:</p>
p_issue_date	IN	DATE		<p>Date Notes receivable was issued (Asia Pacific Requirement)</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_issuer_bank_branch_id	IN	NUMBER(15)		<p>Bank/ Branch issuing the Notes Receivable (Asia Pacific Requirement)</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_cr_id	OUT	NUMBER(15)	Yes	The cash receipt id of the receipt created by the API call.
p_default_site_use	IN	VARCHAR2	No	<p>Indicates if you want to default the site use from p_customer_site_use_id.</p> <p>The default value is Y. Pass N to default nothing.</p> <p>If the Require Billing Location for Receipts system option is selected, then no value is required here.</p>
p_payment_trxn_extension_id				Payment transaction extension identifier
p_org_id				

Parameter	Type	Data-type	Required	Description
p_installment	IN	NUMBER(15)		<p>The installment (or term_sequence_number) of the debit item. Used in conjunction with customer_trx_id to derive the applied payment schedule id if not specified.</p> <p>Default: 1, if only one installment exists for the debit item</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. >0; 2. valid installment of transaction.

Validation

This section explains the validation mechanisms for the various parameters of this API which are relatively more complex and could not be explained in the Description column of the preceding table.

Validating Receipt Method ID

The receipt method ID is validated per the following conditions:

- It must be a valid receipt method ID in the AR_RECEIPT_METHOD table.
- Receipt date must lie between the receipt method start date and end date (if not null).
- The creation method code for the receipt class of this particular receipt method ID should be 'AUTOMATIC,' the remit flag ='Y,' and the confirm flag = 'N' or 'MANUAL.'
- At least one remittance bank account associated with this receipt method ID must have either the multi-currency flag set to 'Y' or the same currency as the receipt currency. In addition, this should have a bank account type = 'INTERNAL' and its inactive date (if specified) greater than the receipt_date.

Validating Remittance Bank Account ID

A remittance bank account ID, which is associated with a particular receipt method, is validated after validating the receipt method ID. If the receipt method ID is invalid, then the validation for the remittance bank account ID is not completed. An error message raised for an invalid value is AR_RAPI_INVALID_REMIT_BK_AC_ID.

The remittance bank account ID must:

- Be a valid remittance bank account ID for the current receipt method.
- Have the multi-currency flag set to 'Y' or the same currency as the receipt currency. In addition, this should have a bank account type = 'INTERNAL' and its inactive date (if specified) greater than the receipt_date.

Validating for Duplicate Receipt

If the combination of the receipt_date, receipt_number, and amount on this receipt matches any existing receipts which have not been reversed, then the error message AR_RW_CASH_DUPLICATE_RECEIPT is raised.

Defaulting

This section explains the defaulting mechanisms for the various parameters of this API which are relatively more complex and could not be explained in the Description column of the preceding table.

Defaulting the Remittance Bank Account ID

In addition to being defaulted from the remittance bank account name and/or remittance bank account number, the remittance bank account identifier is defaulted from the receipt method that is specified for the cash receipt. If only one remittance bank account is associated with the specified receipt method that has the multi-currency flag = 'Y' or has same currency as the receipt currency, and the receipt date is within its start date and end date range, then that remittance bank account is used as the default value.

Example

Objective:

To create an identified cash receipt using a call to *Ar_receipt_api_pub.Create_cash* and passing a minimum number of input parameters.

This table lists the entered parameters:

Parameter	Entered Value	Default Value
p_api_version	1.0	
p_init_msg_list	FND_API.G_TRUE	
p_receipt_number	'aj_test_api_1'	
p_amount	1000	

Parameter	Entered Value	Default Value
p_receipt_method_id	1001	
p_customer_name	'Computer Service and Rentals'	

This table lists the defaulted input parameters, which were not entered:

Parameter	Entered Value	Default Value
p_customer_id		1006
p_currency_code		USD
p_receipt_date		10-FEB-2000
p_gl_date		10-FEB-2000
p_deposit_date		10-FEB-2000
p_customer_site_use_id		1025
p_override_remit_bank_account_flag		'Y'
p_remittance_bank_account_id		10001
p_maturity_date		10-FEB-2000

The API call in this case would be:

```
Ar_receipt_api_pub.Create_cash(
    p_api_version      => 1.0,
    p_init_msg_list   => FND_API.G_TRUE,
    p_receipt_number   => 'aj_test_api_1',
    p_amount           => 1000,
    p_receipt_method_id => 1001,
    p_customer_name    => 'Computer Service and Rentals',
    p_cr_id            => l_cr_id,
    x_return_status     => l_return_status,
    x_msg_count         => l_msg_count,
    x_msg_data          => l_msg_data);
```

The warnings and the error messages that the API puts on the message stack are

retrieved after execution of this API by the calling program in the following manner:

```
IF l_msg_count = 1 Then
    --there is one message raised by the API, so it has been sent out
    --in the parameter x_msg_data, get it.
    l_msg_data_out := l_msg_data;
ELSIF l_msg_count > 1 Then
    --the messages on the stack are more than one so call them in a loop
    -- and put the messages in a PL/SQL table.
    loop
        count := count +1 ;
        l_mesg := FND_MSG_PUB.Get;
        If l_mesg IS NULL Then
            EXIT;
        else
            Mesg_tbl(count).message := l_mesg;
        End if;
    end loop;
END IF;
```

Depending on the message level threshold set by the profile option FND_API_MSG_LEVEL_THRESHOLD, the messages put on the message stack may contain both the error messages and the warnings.

Result:

We were able to create an identified cash receipt by specifying only six input parameters in our call to this API.

Similarly, without initializing the message stack (p_init_msg_list not passed and defaulted), you can create an unidentified cash receipt (without a customer) by passing only four input parameters to this API call.

Ar_receipt_api_pub.Apply

Call this routine to apply the cash receipts of a customer (identified cash receipt) to a debit item. This debit item could be of the same customer or related customer, or an unrelated customer, depending on the value of the Allow Payment of Unrelated Transactions system option. This API routine has 3 output and 34 input parameters in total. Based on the type, the following is the breakdown of the parameters:

Input

Standard API parameters: 4

Application parameters: 28 + 1 (descriptive flexfield record parameters)

+ 1 (global descriptive flexfield record parameters)

Output

Standard API parameters: 3

Application parameters: 0

Parameter Descriptions

For a description of this routine's standard parameters, see [Ar_receipt_api_pub.Create_cash](#), page 8-3.

The following table lists the parameters that pertain specifically to the Apply routine.

Note: If required parameters are not passed in a call to this API, then the call will fail. However, depending on the business scenario, you will have to pass in values for other parameters to successfully create the business object. Otherwise, error messages will be reported.

Parameter	Type	Data-type	Required	Description
p_cash_receipt_id	IN	NUMBER(15)		<p>The cash_receipt_id of the receipt which needs to be applied to a given debit item.</p> <p>Default: None</p> <p>Validation:</p> <ol style="list-style-type: none">1. Type must be 'CASH'2. Status must not be Reversed or Approved3. The receipt must not be Unidentified <p>Error: AR_RAPI_CASH_RCPT_ID_INVALID AR_RAPI_CASH_RCPT_ID_NULL</p>
p_receipt_number	IN	VARCHAR2(30)		<p>The receipt number of the receipt to be applied. Used to default the cash_receipt_id.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPI_RCPT_NUM_INVALID</p>
p_customer_trx_id	IN	NUMBER(15)		<p>The customer_trx_id of the debit item to which the receipt is to be applied</p> <p>Default: None</p> <p>Validation: Detailed in Validation, page 8-30</p> <p>Error: Detailed in Validation, page 8-30</p>

Parameter	Type	Data-type	Required	Description
p_trx_number	IN	VARCHAR2(20)		<p>The trx_number of the debit item to which the receipt is to be applied. Used to default the customer_trx_id</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPI_TRX_NUM_INVALID</p>
p_installment	IN	NUMBER(15)		<p>The installment (or term_sequence_number) of the debit item. Used in conjunction with customer_trx_id to derive the applied payment schedule id if not specified.</p> <p>Default: 1, if only one installment exists for the debit item</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. >0; 2. valid installment of transaction. <p>Also see Validation, page 8-30</p> <p>Error: AR_RAPI_INSTALL_NULL</p>
p_applied_payment _schedule_id	IN	NUMBER(15)		<p>The payment schedule id of the debit item. Also used to derive the customer_trx_id if not specified</p> <p>Default: Defaulted based on the installment and the customer_trx_id</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. > 0 2. It must correspond to Customer trx id and installment specified. 3. It must have the status \leftrightarrow 'CL' if the show closed invoices flag \leftrightarrow 'Y' <p>Error: AR_RAPI_APP_PS_ID_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_amount_applied	IN	NUMBER		<p>The transaction amount to which the receipt is to be applied, in the transaction currency.</p> <p>Default: The default amount applied can be either the open amount of the transaction or the unapplied amount of the receipt, but you can change it. Use the AR: Always Default Transaction Balance for Applications profile option, <i>Oracle Receivables Implementation Guide</i> to control how Receivables defaults the applied amount.</p> <p>The profile option's defaulting rules are:</p> <ul style="list-style-type: none"> • If you set the profile option to <i>Yes</i>, then the default amount applied is the remaining transaction amount. • If you set the profile option to <i>No</i>, or if a null value exists, then the defaulting rule is: <ul style="list-style-type: none"> 1. If the unapplied receipt amount is greater than or equal to the transaction, then the default amount applied is the remaining transaction amount. 2. If the unapplied receipt amount is less than the remaining transaction amount, then the default amount applied is the unapplied receipt amount. 3. If the unapplied receipt amount is negative, then the default amount applied is the remaining transaction amount. <p>Discounts, if applicable, are taken into account by the discounts routine which calculates the amount applied.</p> <p>Validation: Detailed in Validation, page 8-30</p> <p>Error: Detailed in Validation, page 8-30</p>

Parameter	Type	Data-type	Required	Description
p_amount_applied_f rom	IN	NUMBER		<p>The allocated receipt amount in receipt currency.</p> <p>Use only for cross currency receipt applications. Do not use when transaction and receipt currencies are the same.</p> <p>Default:</p> <ul style="list-style-type: none"> For a same currency application, defaults to null For the cross currency application, defaults to $trans_to_receipt_rate * amount_applied$ <p>Validation: Detailed in Validation, page 8-30</p> <p>Error: Detailed in Validation, page 8-30</p>
p_trans_to_receipt_r ate	IN	NUMBER		<p>For cross currency receipts, the exchange rate used to convert an amount from a foreign currency to functional currency</p> <p>Default: Detailed in Defaulting, page 8-29</p> <p>Validation: Detailed in Validation, page 8-30</p> <p>Error: Detailed in Validation, page 8-30</p>
p_discount	IN	NUMBER		<p>Discount on the debit item, entered in the invoice currency</p> <p>Default: Detailed in Defaulting, page 8-29</p> <p>Validation: Detailed in Validation, page 8-30</p> <p>Error: Detailed in Validation, page 8-30</p>

Parameter	Type	Data-type	Required	Description
p_apply_date	IN	DATE		<p>Date the application was applied.</p> <p>Default:</p> <ol style="list-style-type: none"> 1. Receipt date, if receipt date \geq system date 2. System date, if receipt date $<$ system date <p>Validation: apply date \geq transaction date apply date \geq receipt date</p> <p>Error: AR_APPLY_BEFORE_TRANSACTION AR_APPLY_BEFORE_RECEIPT</p>
p_gl_date	IN	DATE		<p>Date that this application will be posted to the General Ledger</p> <p>Default: Detailed in Defaulting, page 8-29</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. Validated as per standard gl date validation described for the gl date in Create_cash routine 2. \geq transaction gl date 3. \geq receipt gl date <p>Error:</p> <p>AR_INVALID_APP_GL_DATE AR_VAL_GL_INV_GL AR_RW_GL_DATE_BEFORE_REC_GL</p>
p_ussql_transaction_code	IN	VARCHAR2(30)		<p>Code defined by public sector accounting.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>

Parameter	Type	Data-type	Required	Description
p_customer_trx_line_id	IN	NUMBER(15)		<p>The customer trx line id of the debit item to which the payment is applied.</p> <p>Default: From the line number if specified</p> <p>Validation: This should be a valid line id for the specified customer trx id.</p> <p>Error: AR_RAPI_TRX_LINE_ID_INVALID</p>
p_line_number	IN	NUMBER		<p>The line number of the debit item to which the payment is applied.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPI_TRX_LINE_NO_INVALID</p>
p_show_closed_invoices	IN	VARCHAR2(1)		<p>This flag decides whether to do the receipt application against closed invoices. The valid values are 'Y' and 'N'</p> <p>Default: 'N'</p> <p>Validation: Any other value is treated as 'N'.</p> <p>Error: None</p>
p_event	IN	VARCHAR2(50)		<p>The event that resulted in the creation of the receipt. Currently used only by Bills Receivables.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_move_deferred_tax	IN	VARCHAR2(1)		<p>Depending on maturity date, this flag indicates when deferred tax should be moved on the accounting event.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>

Parameter	Type	Data-type	Required	Description
p_attribute_record	IN	attribute_rec_type		<p>This is a record type which contains all the 15 descriptive flexfield segments and one descriptive flexfield structure defining column. It represents the Receipt Application Information flexfield.</p> <p>Default: DFF APIs used to do the defaulting and validation</p> <p>Validation: DFF APIs used to do the defaulting and validation</p> <p>Error: AR_RAPI_DESC_FLEX_INVALID</p>
p_global_attribute_record	IN	global_attribute_rec_type		<p>This is a record type which contains all the 20 global descriptive flexfield segments and One global descriptive flexfield structure defining column.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_comments	IN	VARCHAR2(240)		User's comments
p_payment_set_id	IN	NUMBER(15)		<p>Payment set ID is populated only for a prepayment receipt that needs to be applied to a given debit item.</p> <p>Default: None</p> <p>Validation: None</p>
p_application_ref_type	IN	VARCHAR2(30)		<p>Application reference type – this determines the context of the application reference fields.</p> <p>Default: None</p> <p>Validation: Must be Null or, if a Trade Management deduction is being created, then must be 'CLAIM' (Trade Management must be installed).</p> <p>Error: AR_RAPI_INVALID_APP_REF</p>

Parameter	Type	Data-type	Required	Description
p_application_ref_id	IN	NUMBER(15)		Must be NULL.
p_application_ref_n um	IN	VARCHAR2(30)		The reference number relating to the application reference type. If application reference type is 'CLAIM', then this would be a deduction number. Default: None Validation: If populated, then must be an existing deduction number in Trade Management. Error: AR_RAPI_INVALID_CLAIM_NUM
p_secondary_appli cation_ref_id	IN	NUMBER(15)		The secondary application reference ID related to the application reference type. Default: None Validation: If populated, and if the application reference type is 'CLAIM', then this must contain a valid claim ID in Trade Management. Error: AR_RW_INVALID_CLAIM_ID
p_application_ref_re ason	IN	VARCHAR2(30)		The reason code related to the application reference type. Default: None Validation: If populated, and if the application reference type is 'CLAIM', then this must contain a valid reason code ID in Trade Management. Error: AR_RAPI_INVALID_REF_REASON
p_customer_referenc e	IN	VARCHAR2(100)		Reference supplied by customer.

Parameter	Type	Data-type	Required	Description
p_customer_reason	IN	VARCHAR2(30)		<p>Reason code supplied by customer, in the context of an application reference type of 'CLAIM'.</p> <p>Default: None</p> <p>Validation: None in Oracle Receivables (the attempt to match to an Oracle reason code is made in Trade Management).</p>

Defaulting

This section explains the defaulting mechanisms for the various parameters of this API, which are relatively more complex in nature and could not be explained in the Description column of the preceding table.

Trans to receipt rate

For a cross currency application, the transaction to receipt rate is defaulted by the following rules:

- Check if a fixed rate exists (using the GL APIs) between the transaction currency and the receipt currency. If yes, then get it and use it as the default.
- If there is no fixed rate relationship between the transaction currency and the receipt currency, and the profile option AR: Cross Currency Exchange Rate Type has a value, then try to derive a rate from the database using the profile option value and the cash receipt date as the exchange rate date. If you get a rate from the database, then use it as default.
- If the amount_applied and the amount_applied_from are specified, then derive the transaction to receipt rate using the following equation:

$$\text{trans_to_receipt_rate} = \text{amount_applied_from} / \text{amount_applied}$$

GL Date

The application GL date is defaulted to the greater of the Receipt GL date or, depending on the value of the profile option AR: Application GL Date Default, the system date or transaction GL date.

Discount

Defaults to the maximum discount available on the transaction, as of the date of application, which is internally calculated by the discounts routine.

Validation

This section explains the validation mechanisms for the various parameters of this API which are relatively more complex in nature and could not be explained in the Description column of the preceding table.

Customer Trx ID

The customer_trx_id is validated using the conditions mentioned below:

- If the Show Closed Invoices flag is set to 'Y,' then the current transaction + installment can have a payment schedule status of Closed ('CL'). Otherwise, the payment schedule status must be Open ('OP').
- If the Allow Payment of Unrelated Transactions system option = 'Y,' then the current transaction can be for a customer who is not related to the customer on the receipt. Otherwise, the transaction must be for the same or related customer on the receipt.
- The transaction must be an Invoice, Credit Memo, Debit Memo, Deposit, or Chargeback.

Note: This transaction can be in a currency that is different from the receipt currency.

Depending on the specified input parameters, one of the following error messages is raised for an invalid transaction:

- AR_RAPI_TRX_ID_INST_INVALID
- AR_RAPI_TRX_NUM_INST_INVALID
- AR_RAPI_CUST_TRX_ID_INVALID
- AR_RAPI_TRX_NUM_INVALID
- AR_RAPI_APP_PS_ID_INVALID

For details of these messages, refer to [Messages, page 8-137](#).

Amount Applied

- The amount applied cannot be null. The error message raised for an invalid value is AR_RAPI_APPLIED_AMT_NULL.
- The amount applied must not be greater than the line amount for the given customer_trx_line ID (if specified). The error message raised for an invalid value is AR_RW_APPLIED_GREATER_LINE.

- Depending on the creation sign, natural application flag, allow overapplication flag, and the amount due remaining of the specified transaction installment, the amount applied is validated to check for overapplication and natural application. The error messages raised for invalid values are AR_CKAP_OVERAPP, AR_CKAP_NATURALAPP, and AR_CKAP_CT_SIGN. For details of the messages, refer to Messages, page 8-137.
- For a cross currency application, the following equation should always be valid:

$$\text{amount applied} * \text{trans to receipt rate} = \text{amount applied from}$$

The error message raised is AR_RAPI_INVALID_CC_AMTS.

Amount Applied From

- During a cross-currency receipt application, the amount applied from cannot be null. The error message raised for an invalid value is AR_RAPI_AMT_APP_FROM_NULL.
- The amount applied from cannot be greater than the unapplied amount available on the receipt. The error message raised for invalid values is AR_RW_APP_NEG_UNAPP.
- If the transaction currency and the receipt currency are the same, then the amount applied from must always be null. The error message raised for an invalid value is AR_RAPI_AMT_APP_FROM_INVALID.
- As mentioned previously for a cross currency application, the following equation must always be valid:

$$\text{amount applied} * \text{trans to receipt rate} = \text{amount applied from}$$

Trans to Receipt Rate

- For a cross currency application, the trans to receipt rate should have a positive value. The error message raised for an invalid value is AR_RW_CC_RATE_POSITIVE.
- If the transaction currency and the receipt currency are the same, then the rate should not have any value specified. The error message raised for an invalid value is AR_RAPI_INVALID_CC_RATE.
- For a cross currency application, the following equation should always be valid:

$$\text{amount applied} * \text{trans to receipt rate} = \text{amount applied from}$$

If this condition is violated, then the error raised is AR_RAPI_CC_RATE_AMTS_INVALID.

Discount

- If the amount due original on the transaction (debit item) is negative, then discount = 0 or null. The error message raised for an invalid value is AR_RW_NO_DISCNT.
- If amount applied > 0, then the discount cannot be negative. The error message raised for an invalid value is AR_RW_VAL_NEG_DISCNT.
- If partial discount flag = 'N' and the transaction has not been completely paid off by the receipt application, then the discount = 0 or null. The error message raised for an invalid value is AR_NO_PARTIAL_DISC.
- The discount must not be greater than the maximum discount allowed on the transaction, which is internally calculated in the API by the discounts routine. The error message raised for an invalid value is AR_RW_VAL_DISCOUNT.

If the Allow Unearned Discounts system option = 'N,' then the discount must be less than or equal to the allowed earned discount, which gets internally calculated in the API by the discounts routine for the given transaction. The error message raised for an invalid value is AR_RW_VAL_UNEARNED_DISCOUNT.

Application Ref Num

If p_application_ref_type is 'CLAIM', then the application reference number can be populated with a valid deduction number from Trade Management. This deduction/overpayment must be in the same currency as the debit item being applied to. Otherwise, the error message raised is AR_RAPI_INVALID_CLAIM_NUM.

Secondary Application Ref ID

If p_application_ref_type is 'CLAIM', then the secondary application reference ID can be populated with a valid claim ID from Trade Management. This deduction/overpayment must be in the same currency as the debit item being applied to. Otherwise, the error message raised is AR_RAPI_INVALID_CLAIM_NUM.

If both the application reference number and the secondary application reference ID are left null, and p_application_ref_type is 'CLAIM', then a new claim will be created in Trade Management.

Example

Objective:

To apply a cash receipt in functional currency to an invoice in functional currency having only one installment using a call to the API *Ar_receipt_api_pub.Apply* and passing a minimum number of input parameters.

This table lists the entered parameters:

Parameter	Entered Value	Default Value
p_api_version	1.0	
p_trx_number	'aj_test_trx_1'	
p_receipt_number	'aj_test_cr_2'	

This table lists the defaulted input parameters, which were not entered:

Parameter	Entered Value	Default Value
p_customer_trx_id		187807
p_installment		1
p_cash_receipt_id		23927
p_gl_date		10-FEB-2000
p_applied_payment_schedule_id		36271
p_apply_date		10-FEB-2000
p_amount_applied		98
p_amount_applied_from		98
p_discount		2
p_show_closed_invoices		'N'

Result:

We were able to apply the cash receipt against the specified transaction by specifying only three input parameters in our call to this API. The retrieval and handling of the warnings and the error messages, put on the message stack by the API during execution, are the same as described in Defaulting, page 8-18.

Ar_receipt_api_pub.Create_and_apply

Call this routine to create a cash receipt and apply it to a specified installment of a debit item. This debit item could be for the same customer or related customer, or for an unrelated customer, depending on the Allow Payment of Unrelated Transactions system option.

This is essentially a superset of the *ar_receipt_api_pub.Create_cash* and *Ar_receipt_api_pub.Apply* APIs, and contains the same parameters as contained in those two APIs. During the call to this API, if the creation of the receipt is successfully completed but its application to the debit item fails, then the receipt creation is also rolled back.

This routine calls Oracle Payments, where required. See *Integration with Oracle Payments*, page 8-2.

Note: To create credit card receipts that need to be processed by Oracle Payments APIs, you must pass the *p_call_payment_processor* parameter as *fnd_api.g_true*. Additionally, you must specify the *p_customer_bank_account_id* parameter.

This API routine has 3 output and 59 input parameters in total. Based on the type, the following is the breakdown of the parameters:

Input

Standard API parameters: 4

Application parameters: 47 + 2 (descriptive flexfield record parameter)
+ 2 (global descriptive flexfield record parameter)

Output

Standard API parameters: 3

Application parameters: 0

Parameter Descriptions

For a description of this routine's standard parameters, see *Ar_receipt_api_pub.Create_cash*, page 8-3.

The following table lists the parameters that are relevant to the receipt creation and application for the API.

Note: If required parameters are not passed in a call to this API, then the call will fail. However, depending on the business scenario, you will have to pass in values for other parameters to successfully create

the business object. Otherwise, error messages will be reported.

Parameter	Type	Data-type	Required	Description
p_usr_currency_code	IN	VARCHAR2		<p>The translated currency code. Used to derive the p_currency_code if it is not entered.</p> <p>Default: None</p> <p>Validation: Should be a valid currency, so that we can derive the corresponding currency code.</p> <p>Error: AR_RAPI_USR_CURR_CODE_INVALID</p>
p_currency_code	IN	VARCHAR2(15)		<p>The actual currency code that gets stored in AR tables.</p> <p>Default:</p> <ol style="list-style-type: none"> 1. Derived from p_usr_currency_code if entered. Otherwise, 2. Defaulted to the functional currency code. <p>Validation: Validated against the currencies in fnd_currencies table.</p> <p>Error: AR_RAPI_CURR_CODE_INVALID</p> <p>Warning: AR_RAPI_FUNC_CURR_DEFAULTED</p>
p_usr_exchange_rate_type	IN	VARCHAR2		<p>The translated exchange rate type. Used to derive the p_exchange_rate_type if it has not been entered.</p> <p>Default: None</p> <p>Validation: Should be a valid rate type.</p> <p>Error: AR_RAPI_USR_X_RATE_TYP_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_exchange_rate_ty pe	IN	VARCHAR2(30)		<p>Exchange rate type stored in AR tables.</p> <p>Default:</p> <ol style="list-style-type: none"> 1. In case of foreign currency receipt, derived from p_usr_exchange_rate_type 2. If p_usr_exchange_rate_type is null, then defaulted from AR: Default Exchange Rate Type profile option 3. Should be left null, if the receipt is in the same denomination as functional currency <p>Validation: Validated against values in gl_daily_conversion_types table</p> <p>Error: AR_RAPI_X_RATE_TYPE_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_exchange_rate	IN	NUMBER		<p>The exchange rate between the receipt currency and the functional currency.</p> <p>Default:</p> <ol style="list-style-type: none"> 1. Derived from the Daily Rates table for rate_type \neq 'User' in case of non-functional currency 2. If profile option Journals: Display Inverse Rate = 'Y', set user entered value to $1/p_exchange_rate$ 3. The entered value is rounded to a precision of 38. <p>Validation:</p> <ol style="list-style-type: none"> 1. In case of non-functional currency the rate should have a positive value for rate type= 'User' 2. For non-functional currency and type \neq 'User' the user should not specify any value. <p>Error: AR_RAPI_X_RATE_INVALID AR_RAPI_X_RATE_NULL</p>
p_exchange_rate_date	IN	DATE		<p>The date on which the exchange rate is valid.</p> <p>Default: Receipt date</p> <p>Validation: For a non-functional currency and type \neq 'User' there should be a valid rate existing in the database for this date. This is a cross validation of type, currency and date</p> <p>Error: AR_NO_RATE_DATA_FOUND</p>

Parameter	Type	Data-type	Required	Description
p_amount	IN	NUMBER	Yes	<p>The cash receipt amount.</p> <p>Default: Null</p> <p>Validation: >0</p> <p>Error: AR_RAPI_REC_AMT_NEGATIVE AR_RAPI_RCPT_AMOUNT_NULL</p>
p_factor_discount_a mount	IN	NUMBER		<p>The bank charges on the cash receipt.</p> <p>Default: None</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. Bank charges are not allowed if profile option AR: Create Bank Charges = 'No'. 2. Bank charges not allowed if the receipt state, derived from the receipt class of the receipt method, <> 'Cleared'. 3. If allowed, then >= 0. <p>Error: AR_BK_CH_NOT_ALLWD_IF_NOT_CLR AR_JG_BC_AMOUNT_NEGATIVE</p>
p_receipt_number	IN	VARCHAR2(30)		<p>The receipt number of the receipt to be created.</p> <p>Default: If not specified, the receipt number is defaulted from the document sequence value.</p> <p>Validation: Receipt number should not be null</p> <p>Error: AR_RAPI_RCPT_NUM_NULL</p>
p_receipt_date	IN	DATE		<p>The receipt date of the entered cash receipt.</p> <p>Default: System date</p> <p>Validation: None</p> <p>Error: None</p>

Parameter	Type	Data-type	Required	Description
p_gl_date	IN	DATE		<p>Date that this receipt will be posted to the General Ledger.</p> <p>Default: Gets defaulted to the receipt date if it is a valid gl_date.</p> <p>Validation: The date is valid if the following conditions are true:</p> <ul style="list-style-type: none"> • The date is in an Open or Future period • The period cannot be an Adjustment period <p>If the date is invalid, then:</p> <ul style="list-style-type: none"> • If the most recent open period is prior to the receipt date: last date of that period • If there is a period open after the receipt date: first date of the last open period <p>Error: AR_INVALID_APP_GL_DATE</p>
p_maturity_date	IN	DATE		<p>Receipt maturity date.</p> <p>Default: Deposit date</p> <p>Validation: \geq p_receipt_date</p> <p>Error: AR_RW_MAT_BEFORE_RCT_DATE</p>
p_customer_id	IN	NUMBER(15)		<p>The customer_id for the paying customer.</p> <p>Default: Refer to Defaulting, page 8-52</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. Customer exists and has prospect code = 'CUSTOMER' 2. Customer has a profile defined at customer level <p>Error: AR_RAPI_CUST_ID_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_customer_name	IN	VARCHAR2(50)		<p>The name for the entered customer. Used to default the customer id if not specified.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPI_CUS_NAME_INVALID</p>
p_customer_number	IN	VARCHAR2(30)		<p>The customer number. Used to default the customer_id if not specified.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPI_CUS_NUM_INVALID</p>
p_customer_bank_ac count_id	IN	NUMBER(15)		<p>The customer bank account ID.</p> <p>Default: From bank account ID/number.</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. It must be a valid bank account of the paying customer. 2. The inactive date (if defined) of the bank account should be greater than the receipt_date. 3. The receipt date must be within the Start date and the End date of the bank account uses. <p>Error: AR_RAPI_CUS_BK_AC_2_INVALID AR_RAPI_CUS_BK_AC_ID_INVALID</p>
p_customer_bank_ac count_num	IN	VARCHAR2(30)		<p>The customer bank account number. Used to default the customer bank account id, if not specified.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>

Parameter	Type	Data-type	Required	Description
p_customer_bank_ac count_name	IN	VARCHAR2(80)		<p>The customer bank account name. Used to default the customer bank account id, if not specified.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_customer_location	IN	VARCHAR2(40)		<p>The Bill_To location for the customer. Used to derive the p_customer_site_use_id.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPI_CUS_LOC_INVALID</p>
p_customer_site_use _id	IN	NUMBER(15)		<p>The Bill_To site_use_id for the customer.</p> <p>Default:</p> <ol style="list-style-type: none"> 1. Defaulted from customer location. Otherwise, <ol style="list-style-type: none"> 2. Primary Bill_To customer site_use_id of the customer. <p>Validation: It should be a valid Bill_To site of the paying customer.</p> <p>Error:</p> <p>AR_RAPI_CUS_SITE_USE_ID_INVALID</p>
p_customer_receipt_ reference	IN	VARCHAR2(30)		<p>This column is used to store a customer receipt reference value that the customer supplies at the confirmation time.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>

Parameter	Type	Data-type	Required	Description
p_override_remit_ba nk_account_flag	IN	VARCHAR2(1)		<p>The flag value decides when the remittance bank account can be overridden by the remittance selection process.</p> <p>Default: 'Y'</p> <p>Validation: valid values 'Y' and 'N'</p> <p>Error: AR_RAPI_INVALID_OR_REMIT_BK_AC</p>
p_remittance_bank_ account_id	IN	NUMBER(15)		<p>Identifies the user's bank account for depositing the receipt.</p> <p>Default:</p> <ol style="list-style-type: none"> 1. From remittance bank account number 2. From the receipt method based on logic mentioned in Defaulting, page 8-18 <p>Validation: Validation logic detailed in Validation, page 8-17</p> <p>Error: AR_RAPI_Rem_BK_AC_ID_INVALID AR_RAPI_Rem_BK_AC_ID_NULL</p>
p_remittance_bank_ account_num	IN	VARCHAR2(30)		<p>The remittance bank account number. Used to default the remittance bank account id if not specified.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPI_Rem_BK_AC_NUM_INVALID</p>
p_remittance_bank_ account_name	IN	VARCHAR2(50)		<p>The remittance bank account name. Used to default the remittance bank account id if not specified.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPI_Rem_BK_AC_NAME_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_deposit_date	IN	DATE		<p>The deposit date.</p> <p>Default: receipt date</p> <p>Validation: None</p> <p>Error: None</p>
p_receipt_method_id	IN	NUMBER(15)		<p>Identifies the receipt method of the receipt.</p> <p>Default: From receipt method name</p> <p>Validation: Validation detailed in Validation, page 8-17</p> <p>Error: AR_RAPI_INVALID_RCT_MD_ID</p>
p_receipt_method_name	IN	VARCHAR2(30)		<p>The receipt method name of the receipt.</p> <p>Used to default the receipt method id if not specified.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p> <p>Note: To use credit card refund functionality, ensure that remittance of the original receipt is performed within Oracle Receivables. Do this by setting the remittance method on the receipt method's associated receipt class to <i>Standard</i>.</p> <p>Warning: If you use this API to both authorize and capture credit card payments, then set the remittance method to <i>None</i>. Note, however, that with this setting, you cannot use standard credit card refund functionality. Instead, you must refund such payments <i>outside</i> Receivables.</p>

Parameter	Type	Data-type	Required	Description
p_doc_sequence_value	IN	NUMBER		<p>Value assigned to document receipt.</p> <p>Default: Detailed in Defaulting, page 8-18.</p> <p>Validation:</p> <ul style="list-style-type: none"> • You should not pass a value, if the current document sequence is automatic. • Document sequence value should not be entered if profile option Sequential Numbering is set to Not Used. <p>Error: AR_RAPI_DOC_SEQ_AUTOMATIC AR_RAPI_DOC_SEQ_VAL_INVALID</p>
p_ussgl_transaction_code	IN	VARCHAR2(30)		<p>Code defined by public sector accounting.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_anticipated_clearing_date	IN	DATE		<p>Date the receipt is expected to be cleared.</p> <p>Default: None</p> <p>Validation: \geq gl_date</p> <p>Error: AR_RW_EFFECTIVE_BEFORE_GL_DATE</p>
p_event	IN	VARCHAR2		<p>The event that resulted in the creation of the receipt. Currently used only by Bills Receivables.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>

Parameter	Type	Data-type	Required	Description
p_called_from	IN	VARCHAR2(20)		<p>This parameter is used to identify the calling routine. Currently used to identify only the 'BR_REMIT' program.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_attribute_record	IN	attribute_rec_type		<p>This is a record type which contains all the 15 descriptive flexfield segments and one descriptive flexfield structure defining column. It represents the Receipt Information flexfield.</p> <p>Default: DFF APIs used to do the defaulting and validation</p> <p>Validation: DFF APIs used to do the defaulting and validation</p> <p>Error: AR_RAPI_DESC_FLEX_INVALID</p>
p_global_attribute_record	IN	global_attribute_record_type		<p>This is a record type which contains all the 20 global descriptive flexfield segments and one global descriptive flexfield structure defining column.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error:</p>
p_issuer_name	IN	VARCHAR2(50)		<p>Issuer name of Notes Receivable (Asia Pacific Requirement).</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>

Parameter	Type	Data-type	Required	Description
p_issue_date	IN	DATE		<p>Date when the note receivable was issued (Asia Pacific Requirement).</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_customer_trx_id	IN	NUMBER(15)		<p>The customer_trx_id of the debit item to which the receipt is to be applied.</p> <p>Default: None</p> <p>Validation: Detailed in Validation, page 8-30</p> <p>Error: Detailed in Validation, page 8-30</p>
p_trx_number	IN	VARCHAR2(20)		<p>The trx_number of the debit item to which the receipt is to be applied. Used to default the customer_trx_id.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPI_TRX_NUM_INVALID</p>
p_installment	IN	NUMBER(15)		<p>The installment (or term_sequence_number) of the debit item. Used in conjunction with customer_trx_id to derive the applied payment schedule id if not specified.</p> <p>Default: 1, if only one installment exists for the debit item</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. >0 2. valid installment of transaction. <p>Also see Validation, page 8-30</p> <p>Error: AR_RAPI_INSTALL_NULL</p>

Parameter	Type	Data-type	Required	Description
p_applied_payment_schedule_id	IN	NUMBER(15)		<p>The payment schedule id of the debit item. Also used to derive the customer_trx_id if not specified.</p> <p>Default: Defaulted based on the installment and the customer_trx_id</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. > 0 2. It must correspond to Customer trx id and installment specified. 3. It must have the status \leftrightarrow 'CL' if the show closed invoices flag \leftrightarrow 'Y' <p>Error: AR_RAPI_APP_PS_ID_INVALID</p>
p_amount_applied	IN	NUMBER		<p>The transaction amount to which the receipt is to be applied. This in the transaction currency.</p> <p>Default: The default amount applied can be either the open amount of the transaction or the unapplied amount of the receipt, but you can change it. Use the AR: Always Default Transaction Balance for Applications profile option, <i>Oracle Receivables Implementation Guide</i> to control how Receivables defaults the applied amount.</p> <p>For the profile option's defaulting rules, see Ar_receipt_api_pub.Apply, page 8-20.</p> <p>Discounts, if applicable, are taken into account by the discounts routine which calculates the amount applied.</p> <p>Validation: Detailed in Validation, page 8-30</p> <p>Error: Detailed in Validation, page 8-30.</p>

Parameter	Type	Data-type	Required	Description
p_amount_applied_f rom	IN	NUMBER		<p>The allocated receipt amount in receipt currency.</p> <p>Use only for cross currency receipt applications. Do not use when transaction and receipt currencies are the same.</p> <p>Default:</p> <ul style="list-style-type: none"> For a same currency application, defaults to null. For the cross currency application, defaults to <code>trans_to_receipt_rate * amount_applied</code>. <p>Validation: Detailed in Validation, page 8-30</p> <p>Error: Detailed in Validation, page 8-30.</p>
p_trans_to_receipt_r ate	IN	NUMBER		<p>For cross currency receipts, the exchange rate used to convert an amount from a foreign currency to functional currency.</p> <p>Default: Detailed in Defaulting, page 8-29</p> <p>Validation: Detailed in Validation, page 8-30</p> <p>Error: Detailed in Validation, page 8-30</p>
p_discount	IN	NUMBER		<p>Discount on the debit item, entered in the invoice currency.</p> <p>Default: Detailed in Defaulting, page 8-29</p> <p>Validation: Detailed in Validation, page 8-30</p> <p>Error: Detailed in Validation, page 8-30</p>

Parameter	Type	Data-type	Required	Description
p_apply_date	IN	DATE		<p>Date the application was applied.</p> <p>Default:</p> <ol style="list-style-type: none"> 1. Receipt date, if receipt date \geq system date 2. System date, if receipt date $<$ system date <p>Validation: apply date \geq transaction date apply date \geq receipt date</p> <p>Error: AR_APPLY_BEFORE_TRANSACTION AR_APPLY_BEFORE_RECEIPT</p>
p_apply_gl_date	IN	DATE		<p>Date that this application will be posted to the General Ledger.</p> <p>Default: Detailed in Defaulting, page 8-29</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. Validated as per standard gl date validation described for the gl date in Create_cash routine 2. Greater than or equal to transaction gl date 3. Greater than or equal to receipt gl date <p>Error: AR_INVALID_APP_GL_DATE AR_VAL_GL_INV_GL AR_RW_GL_DATE_BEFORE_REC_GL</p>
p_app_ussgl_transac tion_code	IN	VARCHAR2(30)		<p>Code defined by public sector accounting.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error:</p>

Parameter	Type	Data-type	Required	Description
p_customer_trx_line_id	IN	NUMBER(15)		<p>The customer trx line id of the debit item to which the payment is applied.</p> <p>Default: From the line number if specified</p> <p>Validation: This should be a valid line id for the specified customer trx id.</p> <p>Error: AR_RAPI_TRX_LINE_ID_INVALID</p>
p_line_number	IN	NUMBER		<p>The line number of the debit item to which the payment is applied.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPI_TRX_LINE_NO_INVALID</p>
p_show_closed_invoices	IN	VARCHAR2(1)		<p>This flag decides whether to do the receipt application against closed invoices. The valid values are 'Y' and 'N'.</p> <p>Default: 'N'</p> <p>Validation: Check for the valid values.</p> <p>Error: AR_RAPI_INVALID_SHOW_CL_INV</p>
p_event	IN	VARCHAR2(50)		<p>The event that resulted in the creation of the receipt. Currently used only by Bills Receivables.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_move_deferred_ta_x	IN	VARCHAR2(1)		<p>Depending on maturity date, this flag indicates when deferred tax should be moved on the accounting event.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>

Parameter	Type	Data-type	Required	Description
p_app_attribute_rec	IN	attribute_rec_type		<p>This is a record type which contains all the 15 descriptive flexfield segments and one descriptive flexfield structure defining column. It represents the Receipt Application Information flexfield.</p> <p>Default: DFF APIs used to do the defaulting and validation</p> <p>Validation: DFF APIs used to do the defaulting and validation</p> <p>Error: AR_RAPI_DESC_FLEX_INVALID</p>
p_app_global_attribute_record	IN	global_attribute_rec_type		<p>This is a record type which contains all the 20 global descriptive flexfield segments and one global descriptive flexfield structure defining column.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error:</p>
p_comments	IN	VARCHAR2(240)		User's comments for the application.
p_call_payment_processor	IN	VARCHAR2 (1)	FND_API.G_FALSE	This is the payment processing indicator flag. Pass as FND_API.G_TRUE, if you want to call Oracle Payments for credit card processing.
p_default_site_use	IN	VARCHAR2	No	<p>Indicates if you want to default the site use from p_customer_site_use_id.</p> <p>The default value is Y. Pass N to default nothing.</p> <p>If the Require Billing Location for Receipts system option is selected, then no value is required here.</p>
p_payment_trxn_extension_id				Payment transaction extension identifier
p_org_id				

Defaulting

This section explains the defaulting mechanisms for the various parameters of this API which are relatively more complex in nature and could not be explained in the Description column of the preceding table.

Customer ID

The p_customer_id is required for the create_and_apply routine because an unidentified receipt cannot be applied to a transaction. If not specified, then the customer ID gets defaulted from one of the following:

- Customer number, customer name, or both
- Bill_to customer on the transaction or drawee customer on the bill (for receipt application against a bill)

If the customer ID is not defaulted by one of the above, then the AR_RAPI_CUST_ID_NULL error is raised.

Example

Objective:

To create a cash receipt in the functional currency against an invoice in USD having only one installment, using a call to the API *Ar_receipt_api_pub.Create_and_Apply* and passing a minimum number of input parameters.

This table lists the entered parameters:

Parameter	Entered Value	Default Value
p_api_version	1.0	
p_receipt_number	'aj_test_api_3'	
p_amount	1000	
p_receipt_method_id	1001	
p_customer_name	'Computer Service and Rentals'	
p_trx_number	'aj_test_trx_3'	

This table lists the defaulted input parameters, which were not entered:

Parameter	Entered Value	Default Value
p_customer_id	1006	
p_currency_code	USD	
p_receipt_date	10-FEB-2000	
p_gl_date	10-FEB-2000	
p_deposit_date	10-FEB-2000	
p_customer_site_use_id	1025	
p_override_remit_bank_account_flag	'Y'	
p_remittance_bank_account_id	10001	
p_maturity_date	10-FEB-2000	
p_customer_trx_id	187809	
p_installment	1	
p_apply_gl_date	10-FEB-2000	
p_applied_payment_schedule_id	36277	
p_apply_date	10-FEB-2000	
p_amount_applied	1000	
p_amount_applied_from	1000	
p_discount	0	
p_show_closed_invoices	'N'	

Result:

We were able to create the cash receipt 'aj_test_api_3' and then apply it against the

invoice 'aj_test_trx_3' by specifying only six input parameters in our call to this API. Both the receipt and the invoice are in the functional currency. The retrieval and handling of the warnings and the error messages, put on the message stack by the API during execution, are the same as described in Defaulting, page 8-18.

Ar_receipt_api_pub.Unapply

Call this routine to unapply a cash receipt application against a specified installment of a debit item or payment_schedule_id. This API routine has 3 output and 14 input parameters in total. Based on the type, the following is the breakdown of the parameters:

Input

Standard API parameters: 4

Application parameters: 10

Output

Standard API parameters: 3

Application parameters: 0

Parameter Descriptions

For a description of this routine's standard parameters, see [Ar_receipt_api_pub.Create_cash, page 8-3](#).

The following table lists the parameters that are specific to the unapplication for the API.

Note: If required parameters are not passed in a call to this API, then the call will fail. However, depending on the business scenario, you will have to pass in values for other parameters to successfully create the business object. Otherwise, error messages will be reported.

Parameter	Type	Data-type	Required	Description
p_cash_receipt_id	IN	NUMBER(15)		<p>The cash_receipt_id of the receipt whose application has to be unapplied.</p> <p>Default: None</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. Status must not be Reversed or Approved 2. The receipt should have an application on it. <p>Error: AR_RAPI_CASH_RCPT_ID_NULL</p>
p_receipt_number	IN	VARCHAR2(30)		<p>The receipt number of the receipt whose application is to be unapplied. Used to default the cash_receipt_id.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPI_RCPT_NUM_INVALID AR_RAPI_TRX_NUM_INST_INVALID</p>
p_customer_trx_id	IN	NUMBER(15)		<p>The customer_trx_id of the debit item against which the specified receipt has an application.</p> <p>Default: None</p> <p>Validation: The transaction must have an application against the specified receipt.</p> <p>Error: AR_RAPI_CUST_TRX_ID_INVALID AR_RAPI_TRX_ID_INST_INVALID</p>
p_trx_number	IN	VARCHAR2(20)		<p>The trx_number of the debit item against which the specified receipt has an application. Used to default the customer_trx_id.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>

Parameter	Type	Data-type	Required	Description
p_installment	IN	NUMBER(15)		<p>The installment (or term_sequence_number) of the debit item. Used in conjunction with customer_trx_id to derive the applied payment schedule id if not specified.</p> <p>Default: 1, if only one installment exists for the debit item</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. >0 2. valid installment of transaction <p>Error: AR_RAPI_INSTALL_NULL AR_RAPI_TRX_ID_INST_INVALID AR_RAPI_TRX_NUM_INST_INVALID</p>
p_applied_payment_schedule_id	IN	NUMBER(15)		<p>The payment schedule id of the debit item. Also used to derive the customer_trx_id, if not specified.</p> <p>Default: Derived from the installment and the customer_trx_id.</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. > 0 2. It must correspond to Customer trx id and installment, if specified. 3. For applications with Bills Receivables installed, you cannot unapply a bill that is in the process of remittance. <p>Error: AR_RAPI_APP_PS_ID_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_receivable_application_id	IN	NUMBER(15)		<p>Identifies the receivable application. Used to derive the customer_trx_id, cash_receipt_id, and the applied_payment_schedule_id, if not specified.</p> <p>Default: Defaulted from the specified transaction and the receipt.</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. Application type must be 'CASH'. 2. Display flag = 'Y' (latest application). 3. The applied payment schedule id of the receivable application record must correspond to the p_applied_payment_schedules_id, if specified. 4. The cash receipt id must correspond to the cash receipt id specified. 5. For applications with Bills Receivables installed, you cannot unapply the application of a bill that is in the process of remittance. <p>Error: AR_RAPI_REC_APP_ID_NULL AR_RAPI_REC_APP_ID_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_reversal_gl_date	IN	DATE		<p>The reversal gl date.</p> <p>Default: Gets defaulted to the application gl date if it is a valid gl_date.</p> <p>Validation:</p> <p>It is valid if the following conditions are true:</p> <ul style="list-style-type: none"> • The date is in an Open or Future period. • The period cannot be an Adjustment period. • The reversal GL date \geq application GL date. • The reversal GL date \geq receipt GL date. <p>If the date is invalid, then:</p> <ul style="list-style-type: none"> • If the most recent open period is prior to the receipt date: last date of that period • If there is a period open after the receipt date: first date of the last open period <p>Error: AR_INVALID_APP_GL_DATE AR_RW_BEFORE_APP_GL_DATE AR_RW_BEFORE_RECEIPT_GL_DATE</p>
p_called_from	IN	VARCHAR2(20)		<p>This parameter is used to identify the calling routine.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_cancel_claim_flag	IN	VARCHAR2(1)		Not used – leave null.

Defaulting

This section explains the defaulting mechanisms for the various parameters of this API which are relatively more complex and could not be explained in the Description column of the preceding table.

Receivable Application ID

If not specified, then the receivable application ID can be defaulted by one of the following:

- Using the specified installment and p_customer_trx_id (derived from p_trx_number if not specified) and p_cash_receipt_id (derived from the receipt number if not specified).
- Using the specified value of p_applied_payment_schedule_id and p_cash_receipt_id (derived from the receipt number if not specified).

Validation

This section explains the cross validations for the various parameters of this API which are relatively more complex and could not be explained in the Description column of the preceding table.

Cross validation between customer_trx_id, applied_payment_schedule_id, cash_receipt_id, and receivable_application_id

- If p_customer_trx_id, p_installment, and p_applied_payment_schedule_id are specified and the two do not point to the same transaction, then the error AR_RAPI_TRX_PS_ID_X_INVALID is raised.
- If the combination of the specified p_applied_payment_schedule_id (or derived from the p_customer_trx_id and p_installment) and the specified p_receivable_application_id is invalid, then the error AR_RAPI_APP_PS_RA_ID_X_INVALID or AR_RAPI_TRX_RA_ID_X_INVALID is raised, depending on the input parameters.

Example

Objective:

To unapply the receipt application against an invoice using the call to API *Ar_receipt_api_pub.Unapply* and passing a minimum number of input parameters.

This table lists the entered parameters:

Parameter	Entered Value	Default Value
p_api_version	1.0	
p_receipt_number	'aj_test_api_4'	

Parameter	Entered Value	Default Value
p_applied_payment_schedule_id	1001	

This table lists the defaulted input parameters, which were not entered:

Parameter	Entered Value	Default Value
p_cash_receipt_id		1006
p_customer_trx_id		USD
p_reversal_gl_date		10-FEB-2000
p_receivable_application_id		29711

The retrieval and handling of the warnings and the error messages, put on the message stack by the API during execution, are the same as described in Defaulting, page 8-18.

Ar_receipt_api_pub.Apply_on_account

Call this routine to apply an on-account application of the specified cash receipt. This API routine has 3 output and 21 input parameters in total. Based on the type, the following is the breakdown of the parameters:

Input

Standard API parameters: 4

Application parameters: 14 + 1 (descriptive flexfield record type)

+ 1 (global descriptive flexfield record type)

Output

Standard API parameters: 3

Application parameters: 0

Parameter Descriptions

For a description of this routine's standard parameters, see Ar_receipt_api_pub.Create_cash, page 8-3.

The following table lists the descriptions of the on-account application-related

parameters of the API.

Note: If required parameters are not passed in a call to this API, then the call will fail. However, depending on the business scenario, you will have to pass in values for other parameters to successfully create the business object. Otherwise, error messages will be reported.

Parameter	Type	Data-type	Required	Description
p_cash_receipt_id	IN	NUMBER(15)		<p>The cash_receipt_id of the receipt which is to be applied on account.</p> <p>Default: None</p> <p>Validation:</p> <ol style="list-style-type: none">1. Type must be 'CASH'2. Status must not be Reversed or Approved <p>The receipt must not be Unidentified</p> <p>Error:</p> <p>AR_RAPI_CASH_RCPT_ID_INVALID</p> <p>AR_RAPI_CASH_RCPT_ID_NULL</p>
p_receipt_number	IN	VARCHAR2(30)		<p>The receipt number of the receipt to be applied on account. Used to default the cash_receipt_id.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error:</p> <p>AR_RAPI_RCPT_NUM_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_amount_applied	IN	NUMBER		<p>The amount on the cash receipt that is to be applied on account.</p> <p>Default: Amount due remaining on the receipt.</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. Greater than or equal to 0. 2. Less than or equal to the amount due remaining on the receipt. <p>Error:</p> <p>AR_RAPI_APPLIED_AMT_NULL</p> <p>AR_RW_APP_NEG_UNAPP</p> <p>AR_RW_AMOUNT_LESS_THAN_AP P</p>
p_apply_date	IN	DATE		<p>Date the application was applied.</p> <p>Default:</p> <ol style="list-style-type: none"> 1. Receipt date, if receipt date \geq system date 2. System date, if receipt date $<$ system date <p>Validation: apply date \geq receipt date</p> <p>Error: AR_APPLY_BEFORE_RECEIPT</p>

Parameter	Type	Data-type	Required	Description
p_apply_gl_date	IN	DATE		<p>Date that this application will be posted to the General Ledger.</p> <p>Default: Defaulted to greater of the receipt date and the system date.</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. Validated as per standard gl date validation described for the gl date in Create_cash routine. 2. \geq receipt gl date. <p>Error:</p> <p>AR_INVALID_APP_GL_DATE</p> <p>AR_RW_GL_DATE_BEFORE_REC_GL</p>
p_ussgl_transaction_code	IN	VARCHAR2(30)		<p>Code defined by public sector accounting.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_attribute_rec	IN	attribute_rec_type		<p>This is a record type which contains all the 15 descriptive flexfield segments and one descriptive flexfield structure defining column. It represents the Receipt Application Information flexfield.</p> <p>Default: DFF APIs used to do the defaulting and validation</p> <p>Validation: DFF APIs used to do the defaulting and validation</p> <p>Error:</p> <p>AR_RAPI_DESC_FLEX_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_global_attribute_r_ec	IN	global_attribute_rec_type		<p>This is a record type which contains all the global descriptive flexfields: One global descriptive flexfield structure defining column and 20 segments.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_comments	IN	VARCHAR2(240)		User comments.
p_application_ref_n_um	IN	VARCHAR2(30)		Deduction number, if resulting from Trade Management claim settlement.
p_secondary_application_ref_id	IN	NUMBER(15)		Claim ID, if resulting from Trade Management claim settlement.
p_customer_reference	IN	VARCHAR2(100)		Reference supplied by customer.
p_called_from	IN	VARCHAR2(20)		<p>This parameter is used to identify the calling routine.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_customer_reason	IN	VARCHAR2(30)		Reason code supplied by customer.
p_secondary_app_ref_type	IN	VARCHAR2(30)		Used for automated receipt handling. Leave null.
p_secondary_app_ref_num	IN	VARCHAR2(30)		Used for automated receipt handling. Leave null.

Note: With an on-account application, you cannot apply a negative amount, as you can do in a regular application of a receipt to a debit item.

Example

Objective:

To apply a cash receipt in the functional currency to an invoice in the functional currency having only one installment, using a call to the API

Ar_receipt_api_pub.Apply_on_account and passing a minimum number of input parameters.

This table lists the entered parameters:

Parameter	Entered Value	Default Value
p_api_version	1.0	
p_receipt_number	'aj_test_cr_2'	

This table lists the defaulted input parameters, which were not entered:

Parameter	Entered Value	Default Value
p_cash_receipt_id		23927
p_gl_date		01-JUN-2000
p_apply_date		01-JUN-2000
p_amount_applied		100

The retrieval and handling of the warnings and error messages, put on the message stack by the API during execution, are the same as described in Defaulting, page 8-18.

Ar_receipt_api_pub.Unapply_on_account

Call this routine to unapply an on-account application on the specified cash receipt. This API routine has 3 output and 9 input parameters in total. Based on the type, the following is the breakdown of the parameters:

Input

Standard API parameters: 4

Application parameters: 5

Output

Standard API parameters: 3

Application parameters: 0

Parameter Descriptions

For a description of this routine's standard parameters, see [Ar_receipt_api_pub.Create_cash](#), page 8-3.

The following table lists the parameters that are relevant to the on-account unapplication for the API.

Note: If required parameters are not passed in a call to this API, then the call will fail. However, depending on the business scenario, you will have to pass in values for other parameters to successfully create the business object. Otherwise, error messages will be reported.

Parameter	Type	Data-type	Required	Description
p_cash_receipt_id	IN	NUMBER(15)		<p>The cash_receipt_id of the receipt whose application has to be unapplied.</p> <p>Default: None</p> <p>Validation:</p> <ol style="list-style-type: none">1. Status must not be Reversed or Approved.2. The receipt must have an on-account application on it. <p>Error: AR_RAPI_CASH_RCPT_ID_INVALID</p>
p_receipt_number	IN	VARCHAR2(30)		<p>The receipt number of the receipt which is to be unapplied. Used to default the cash_receipt_id.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPI_RCPT_NUM_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_receivable_application_id	IN	NUMBER(15)		<p>Identifies the receivable application. Used to derive the customer trx id, cash_receipt_id and the applied_ps_id, if not specified.</p> <p>Default: Refer to Validation, page 8-73.</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. Application type = 'CASH'. 2. Display flag = 'Y' (latest application) and status = 'ACC'. 3. The applied payment schedule id of the receivable application record must correspond to the p_applied_payment_schedules_id, if specified. 4. The cash receipt id must correspond to the cash receipt id specified. <p>Error: AR_RAPI_REC_APP_ID_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_reversal_gl_date	IN	DATE		<p>The reversal gl date.</p> <p>Default: Gets defaulted to the application gl date if it is a valid gl_date.</p> <p>Validation:</p> <p>It is valid if the following conditions are true:</p> <ul style="list-style-type: none"> • The date is in an Open or Future period. • The period cannot be an Adjustment period. • The reversal GL date \geq application GL date. • The reversal GL date \geq receipt GL date. <p>If the date is invalid, then:</p> <ul style="list-style-type: none"> • If the most recent open period is prior to the receipt date: last date of that period • If there is a period open after the receipt date first date of the last open period <p>Error: AR_INVALID_APP_GL_DATE AR_RW_BEFORE_APP_GL_DATE AR_RW_BEFORE_RECEIPT_GL_DATE</p>
p_org_id				

Defaulting

This section explains the defaulting mechanisms for the various parameters of this API which could not be explained in the Description column of the preceding table.

Receivable Application ID

The value for p_receivable_application_id, if not specified, is defaulted from the p_cash_receipt_id (or p_receipt_number). If the receipt does not have an on-account application, then the error AR_RAPI_CASH_RCPT_ID_INVALID is raised. If there is more than one on-account application on the receipt and the value for

`p_receivable_application_id` has not been specified, then the error `AR_RAPI_MULTIPLE_ON_AC_APP` is raised.

Example

Objective:

To unapply the receipt application using the call to API `Ar_receipt_api_pub.Unapply_on_account` and passing a minimum number of input parameters.

This table lists the entered parameters:

Parameter	Entered Value	Default Value
<code>p_api_version</code>	1.0	
<code>p_receipt_number</code>	'aj_test_api_6'	

This table lists the defaulted input parameters, which were not entered:

Parameter	Entered Value	Default Value
<code>p_cash_receipt_id</code>		20338
<code>p_reversal_gl_date</code>		01-JUN-2000

The retrieval and handling of the warnings and error messages, put on the message stack by the API during execution, are the same as described in Defaulting, page 8-18.

Ar_receipt_api_pub.Reverse

Call this routine to reverse cash as well as miscellaneous receipts. This API routine has 3 output and 15 input parameters in total. Based on the type, the following is the breakdown of the parameters:

Input

Standard API parameters: 4

Application parameters: 12 + 1 (descriptive flexfield record type)

1 (global descriptive flexfield record type)

Output

Standard API parameters: 3

Application parameters: 0

Parameter Descriptions

For a description of this routine's standard parameters, see [Ar_receipt_api_pub.Create_cash, page 8-3](#).

The following table lists the descriptions of the reversal-related parameters of the API.

Note: If required parameters are not passed in a call to this API, then the call will fail. However, depending on the business scenario, you will have to pass in values for other parameters to successfully create the business object. Otherwise, error messages will be reported.

Parameter	Type	Data-type	Required	Description
p_cash_receipt_id	IN	NUMBER(15)		<p>The cash_receipt_id of the receipt which needs to be reversed.</p> <p>Default: None</p> <p>Validation: Detailed in Defaulting, page 8-68.</p> <p>Error:</p>
p_receipt_number	IN	VARCHAR2(30)		<p>The receipt number of the receipt to be reversed. Used to default the cash_receipt_id.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPI_RCPT_NUM_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_reversal_category_code	IN	VARCHAR2(20)		<p>Identifies the reason why the payment entry was reversed.</p> <p>Default: None</p> <p>Validation: Validated against the values in ar_lookups for lookup_type = 'REVERSAL_CATEGORY_TYPE'</p> <p>Error:</p> <p>AR_RAPI_REV_CAT_CD_NULL</p> <p>AR_RAPI_REV_CAT_CD_INVALID</p>
p_reversal_category_name	IN	VARCHAR2(80)		<p>This is the translated lookup meaning for the reversal category code. Used to default the reversal category code if not specified.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error:</p> <p>AR_RAPI_REV_CAT_NAME_INVALID</p>
p_reversal_gl_date	IN	DATE		<p>The General Ledger Date that is used to credit the Account CCID for the reversed receipt.</p> <p>Default: System date</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. Validated as per standard gl date validation described for the gl date in Create_cash routine 2. Greater than or equal to receipt gl date <p>Error:</p> <p>AR_INVALID_APP_GL_DATE</p> <p>AR_RW_BEFORE_RECEIPT_GL_DATE</p>

Parameter	Type	Data-type	Required	Description
p_reversal_date	IN	DATE		<p>Date on which the payment entry reversed</p> <p>Default:</p> <ul style="list-style-type: none"> • System date if system date \geq receipt date, else • Receipt date if receipt date $>$ system date <p>Validation: Greater than or equal to receipt date</p> <p>Error: AR_RW_REV_BEFORE_RCT_DATE</p>
p_reversal_reason_code	IN	VARCHAR2(30)		<p>Indicates the reason for reversing receipt</p> <p>Default: None</p> <p>Validation: Validated against the values in ar_lookups for lookup_type = 'CKAJST_REASON'</p> <p>Error:</p> <p>AR_RAPI_REV_REAS_CD_INVALID AR_RAPI_REV_REAS_CD_NULL</p>
p_reversal_reason_name	IN	VARCHAR2(80)		<p>This is the translated lookup meaning for reversal reason code. Used for defaulting the reversal reason code if not specified.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error:</p> <p>AR_RAPI_REV_REAS_NAME_INVALID</p>
p_reversal_comment	IN	VARCHAR2(240)		Comments regarding reversal

Parameter	Type	Data-type	Required	Description
p_attribute_rec	IN	p_attribute_rec		<p>This is a record type which contains all the descriptive flexfields: One descriptive flexfield structure defining column and 15 segments.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_global_attribute_rec	IN	global_attribute_rec_type		<p>This is a record type which contains all the global descriptive flexfields: One global descriptive flexfield structure defining column and 20 segments.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_cancel_claims_flag	IN	VARCHAR2(1)		Not used. Leave null.
p_called_from	IN	VARCHAR2(20)		<p>This parameter is used to identify the calling routine.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_org_id				

Validation

This section explains the validation mechanisms for the various parameters of this API which are relatively more complex in nature and could not be explained in the Description column of the preceding table.

Cash Receipt ID

We have to validate whether this is a valid cash receipt ID, and whether we can reverse this receipt.

The validation steps are:

- This is a valid value in the database. For an invalid value, the error message AR_RAPI_CASH_RCPT_ID_INVALID is raised.
- Status should not be 'Reversed' for this receipt because you cannot reverse an already reversed receipt. The error message raised for an invalid value is AR_RAPI_CASH_RCPT_ID_INVALID.

The receipt is not standard reversible if any two of the following conditions are true:

- If a chargeback was created against an invoice that is applied to the payment to be reversed.
- If there are any payments, adjustments, credit memos, or chargebacks against the above chargeback records in the AR_PAYMENT_SCHEDULES table.
- If the above chargeback has already been posted to the general ledger.

The AR_RAPI_NON_REVERSIBLE error message is raised for invalid values. In these cases, you can create a debit memo reversal to reverse the receipt. Since the Receipt API does not currently support debit memo reversals, you can manually create them using the Receipts workbench.

Example

Objective:

To reverse a cash receipt using a call to the API *Ar_receipt_api_pub.Reverse* and passing a minimum number of input parameters.

This table lists the entered parameters:

Parameter	Entered Value	Default Value
p_api_version	1.0	
p_receipt_number	'aj_test_cr_7'	
p_reversal_category_code	'NSF'	
p_reversal_reason_code	'PAYMENT REVERSAL'	

This table lists the defaulted input parameters, which were not entered:

Parameter	Entered Value	Default Value
p_cash_receipt_id		20340
p_reversal_date		01-JUN-2000
p_reversal_gl_date		01-JUN-2000

The retrieval and handling of the warnings and error messages, put on the message stack by the API during execution, are the same as described in Defaulting, page 8-18.

Ar_receipt_api_pub.activity_application

Call this routine to do an activity application on a cash receipt. Such applications include Short Term Debit (STD) and Receipt Write-off applications.

This API routine has 4 output and 41 input parameters in total. Based on the type, the following is the breakdown of the parameters:

Input

Standard API parameters: 4

Application parameters: 35 + 1 (descriptive flexfield record type)

1 (global descriptive flexfield record type)

Output

Standard API parameters: 3

Application parameters: 1

Parameter Descriptions

For a description of this routine's standard parameters, see Ar_receipt_api_pub.Create_cash, page 8-3.

The following table lists the descriptions of the activity application-related parameters of the API.

Note: If required parameters are not passed in a call to this API, then the call will fail. However, depending on the business scenario, you will have to pass in values for other parameters to successfully create the business object. Otherwise, error messages will be reported.

Parameter	Type	Data-type	Require d*	Description
p_cash_receipt_id	IN	NUMBER(15)		<p>The cash_receipt_id of the receipt which is to be used for the activity application.</p> <p>Default: None</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. Type must be 'CASH' 2. Status must not be Reversed or Approved 3. The receipt must not be Unidentified <p>Error:</p> <p>AR_RAPI_CASH_RCPT_ID_INVALID AR_RAPI_CASH_RCPT_ID_NULL</p>
p_receipt_number	IN	VARCHAR2(30)		<p>The receipt number of the receipt to be applied. Used to default the cash_receipt_id.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPI_RCPT_NUM_INVALID</p>

Parameter	Type	Data-type	Require d*	Description
p_amount_applied	IN	NUMBER		<p>The amount on the cash receipt that is to be applied against the specified activity.</p> <p>Default: Amount due remaining on the receipt.</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. Greater than or equal to 0. 2. Less than or equal to the amount due remaining on the receipt. 3. If a receipt write-off, then must fall within user and system limits (limits must be set). <p>Error:</p> <p>AR_RAPI_APPLIED_AMT_NULL</p> <p>AR_RW_APP_NEG_UNAPP</p> <p>AR_RW_AMOUNT_LESS_THAN_APP</p> <p>AR_WR_NO_LIMIT</p> <p>AR_WR_USER_LIMIT</p> <p>AR_SYSTEM_WR_NO_LIMIT_SET</p> <p>AR_WR_TOTAL_EXCEED_MAX_AMOUNT</p>
p_applied_payment _schedule_id	IN	NUMBER(15)	Yes	<p>The payment schedule identifier here corresponds to special seeded values, such as -2.</p> <p>Default:</p> <p>Validation: The value should correspond to the special seeded values, such as: -2 (Short Term Debt).</p> <p>Error: AR_RAPI_APP_PS_ID_INVALID</p>

Parameter	Type	Data-type	Require d*	Description
p_link_to_customer_trx_id	IN	NUMBER(15)		<p>The customer_trx_id of the Bill for which the activity (e.g. Short Term Debt) application is being done.</p> <p>Default:</p> <p>Validation: The customer_trx_id should correspond to that of a Bill which has a current status of FACTORED or MATURED_PEND_RISK_ELIMINATION.</p> <p>Error: AR_RAPI_LK_CUS_TRX_ID_INVALID</p>
p_receivables_trx_id	IN	NUMBER(15)		<p>Identifier of the receivables activity.</p> <p>Default: None</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. Valid database value. 2. The activity_type for the receivables_trx_id should be in sync with the applied payment schedule identifier passed in. <p>Error: AR_RAPI_REC_TRX_ID_INVALID AR_RAPI_ACTIVITY_X_INVALID</p>
p_apply_date	IN	DATE		<p>Date the application was applied.</p> <p>Default:</p> <ol style="list-style-type: none"> 1. Receipt date, if receipt date \geq system date. 2. System date, if receipt date $<$ system date. <p>Validation: apply date \geq receipt date</p> <p>Error: AR_APPLY_BEFORE_RECEIPT</p>

Parameter	Type	Data-type	Require d*	Description
p_apply_gl_date	IN	DATE		<p>Date that this application will be posted to the General Ledger.</p> <p>Default: Defaulted to greater of the receipt date and the system date.</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. Validated as per standard GL date validation described for the GL date in Create_cash routine. 2. \geq receipt GL date <p>Error: AR_INVALID_APP_GL_DATE AR_RW_GL_DATE_BEFORE_REC_GL</p>
p_ussgl_transaction_code	IN	VARCHAR2(30)		<p>Code defined by public sector accounting.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_attribute_rec	IN	attribute_rec_type		<p>This is a record type which contains all 15 descriptive flexfield segments and one descriptive flexfield structure defining column. It represents the Receipt Application Information flexfield.</p> <p>Default: DFF APIs used to do the defaulting and validation</p> <p>Validation: DFF APIs used to do the defaulting and validation</p> <p>Error: AR_RAPI_DESC_FLEX_INVALID</p>

Parameter	Type	Data-type	Require d*	Description
p_global_attribute_r_ec	IN	global_attribute_rec_type		<p>This is a record type which contains all the global descriptive flexfields: one global descriptive flexfield structure defining column and 20 segments.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_comments	IN	VARCHAR2 (240)		User's comments for the activity application.
p_application_ref_ty_pe	IN	VARCHAR2(30)		Not used. Leave null.
p_application_ref_id	IN	NUMBER(15)		Not used. Leave null.
p_application_ref_n_um	IN	VARCHAR2(30)		If resulting from a settlement of a claim, then this will contain the deduction number.
p_secondary_application_ref_id	IN	NUMBER(15)		If resulting from a settlement of a claim, then this will contain the claim ID.
p_payment_set_id	IN	NUMBER(15)		<p>Payment set ID is populated only when doing a prepayment activity application on a prepayment receipt.</p> <p>Default: None</p> <p>Validation: None</p>
p_receivable_application_id	OUT	NUMBER(15)		The ID of the resulting activity receivable application.
p_customer_reference	IN	VARCHAR2 (100)		Customer supplied reference.

Parameter	Type	Data-type	Require d*	Description
p_val_writeoff_limit_s_flag	IN	VARCHAR2(1)		<p>Flag to indicate whether user-level write-off limits should apply.</p> <p>Default: Y</p> <p>Validation: None</p> <p>Error: None</p>
p_called_from	IN	VARCHAR2(20)		<p>This parameter is used to identify the calling routine.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_nettied_receipt_flag	IN	VARCHAR2(1)		Used for payment netting. Leave null.
p_nettied_cash_receipt_id	IN	NUMBER(15)		Used for payment netting. Leave null.
p_secondary_app_receipt_type	IN	VARCHAR2(30)		Used for automated receipt handling. Leave null.
p_secondary_app_receipt_number	IN	VARCHAR2(30)		Used for automated receipt handling. Leave null.
p_org_id				
p_pay_group_lookup_code	IN	FND_LOOKUPS.lookup_code%TYPE		
p_pay_alone_flag	IN	VARCHAR2		
p_payment_method_code	IN	ap_invoices.payment_method_code%TYPE		
p_payment_reason_code	IN	ap_invoices.payment_reason_code%TYPE		

Parameter	Type	Data-type	Require d*	Description
p_payment_reason_c	IN	ap_invoices.payment_rea son_comments%TYPE		
p_delivery_channel_	IN	ap_invoices.delivery_cha nnel_code%TYPE		
p_remittance_messa	IN	ap_invoices.remittance_		
ge1		message1%TYPE		
p_remittance_messa	IN	ap_invoices.remittance_		
ge2		message2%TYPE		
p_remittance_messa	IN	ap_invoices.remittance_		
ge3		message3%TYPE		
p_party_id	IN	hz_parties.party_id%TYP E		
p_party_site_id	IN	hz_party_sites.party_site _id%TYPE		
p_bank_account_id	IN	ar_cash_receipts.custome r_bank_account_id%TYP E		

Example

Objective:

To apply a cash receipt in then functional currency to a receipt write-off activity in the functional currency, using a call to the API *Ar_receipt_api_pub.activity_application* and passing a minimum number of input parameters.

This table lists the entered parameters:

Parameter	Entered Value	Default Value
p_api_version	1.0	
p_receipt_number	'aj_test_cr_2'	

Parameter	Entered Value	Default Value
p_receivables_trx_id	1300	
p_applied_payment_schedule_id	-3	

This table lists the defaulted input parameters, which were not entered:

Parameter	Entered Value	Default Value
p_cash_receipt_id		23927
p_gl_date		01-JUN-2000
p_apply_date		01-JUN-2000
p_amount_applied		100

The retrieval and handling of the warnings and error messages, put on the message stack by the API during execution, are the same as described in Defaulting, page 8-18.

Ar_receipt_api_pub.activity_unapplication

Call this routine to do a reversal of an activity application on a cash receipt. Such applications include Short Term Debt and Receipt write-off.

This API routine has 3 output and 10 input parameters in total. Based on the type, the following is the breakdown of the parameters:

Input

Standard API parameters: 4

Application parameters: 6

Output

Standard API parameters: 3

Application parameters: 0

Parameter Descriptions

For a description of this routine's standard parameters, see

Ar_receipt_api_pub.Create_cash, page 8-3.

The following table lists the descriptions of the activity unapplication-related parameters of the API.

Note: If required parameters are not passed in a call to this API, then the call will fail. However, depending on the business scenario, you will have to pass in values for other parameters to successfully create the business object. Otherwise, error messages will be reported.

Parameter	Type	Data-type	Required	Description
p_cash_receipt_id	IN	NUMBER(15)		<p>The cash_receipt_id of the receipt on which the activity application needs to be reversed.</p> <p>Default: None</p> <p>Validation:</p> <ol style="list-style-type: none">1. Type must be 'CASH'2. Status must not be Reversed or Approved3. The receipt must not be Unidentified <p>Error: AR_RAPI_CASH_RCPT_ID_INVALID AR_RAPI_CASH_RCPT_ID_NULL</p>
p_receipt_number	IN	VARCHAR2(30)		<p>The receipt number of the receipt to be reversed. Used to default the cash_receipt_id.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPI_RCPT_NUM_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_receivable_application_id	IN	NUMBER(15)		<p>Identifies the receivable application. Used to derive the customer trx id, cash_receipt_id and the applied_ps_id if not specified.</p> <p>Default: Refer to Validation, page 8-73.</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. Application type = 'CASH'. 2. Display flag = 'Y' (latest application) and status = 'ACTIVITY'. 3. The applied payment schedule id of the receivable application record must correspond to the p_applied_payment_schedule_id, if specified. 4. Must correspond to the cash receipt id specified. <p>Error: AR_RAPI_REC_APP_ID_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_reversal_gl_date	IN	DATE		<p>The reversal GL date.</p> <p>Default: Gets defaulted to the application GL date if it is a valid gl_date.</p> <p>Validation:</p> <p>It is valid if the following conditions are true:</p> <ul style="list-style-type: none"> • The date is in an Open or Future period • The period cannot be an Adjustment period • Reversal GL date \geq application GL date • Reversal GL date \geq receipt GL date <p>If the date is invalid, then:</p> <ul style="list-style-type: none"> • If the most recent open period is prior to the receipt date: last date of that period • If there is a period open after the receipt date: first date of the last open period <p>Error: AR_INVALID_APP_GL_DATE AR_RW_BEFORE_APP_GL_DATE AR_RW_BEFORE_RECEIPT_GL_DATE</p>
p_called_from	IN	VARCHAR2(20)	Yes	<p>Indicates which program is calling this API. For example, the BR_REMIT program would be calling this routine for short term debt applications.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_org_id				

Example

Objective:

To unapply an activity application, using a call to the API

Ar_receipt_api_pub.activity_unapplication and passing minimum number of input parameters.

This table lists the entered parameters:

Parameter	Entered Value	Default Value
p_api_version	1.0	
p_receivable_application_id	10051	
p_called_from	NULL	

This table lists the defaulted input parameters, which were not entered:

Parameter	Entered Value	Default Value
p_cash_receipt_id		20338
p_reversal_gl_date		01-JUN-2000

The retrieval and handling of the warnings and error messages, put on the message stack by the API during execution, are the same as described in Defaulting, page 8-18.

Ar_receipt_api_pub.Create_misc

Call this routine to create a miscellaneous receipt.

Note: This routine does *not* call Oracle Payments directly. See Integration with Oracle Payments, page 8-2.

This API routine has 4 output and 37 input parameters in total. Based on the type, the following is the breakdown of the parameters:

Input

Standard API parameters: 4

Application parameters: 33

Output

Standard API parameters: 3

Application parameters: 1

Parameter Descriptions

The following table lists the standard API parameters, which are common to all the routines in the Receipt API:

Parameter	Type	Data-type	Required	Default Value	Description
p_api_version	IN	NUMBER	Yes		Used to compare version numbers of incoming calls to its current version number.
					Unexpected error is raised if version incompatibility exists.
					In the current version of the API, you should pass in a value of 1.0 for this parameter.
p_init_msg_list	IN	VARCHAR2		FND_API.G_FALSE	Allows API callers to request that the API does initialization of the message list on their behalf.
p_commit	IN	VARCHAR2		FND_API.G_FALSE	Used by API callers to ask the API to commit on their behalf.
p_validation_level	IN	NUMBER		FND_API.G_VALID_LEVEL_FULL	Not to be used currently as this is a public API.
x_return_status	OUT	VARCHAR2			Represents the API overall return status. Detailed in Return Status, page 1-4.
x_msg_count	OUT	NUMBER			Number of messages in the API message list.
x_msg_data	OUT	VARCHAR2			This is the message in encoded format if x_msg_count=1.

The following table lists the parameters that are relevant to the miscellaneous receipt:

Parameter	Type	Data-type	Required	Description
p_usr_currency_code	IN	VARCHAR2		<p>The translated currency code. Used to derive the p_currency_code if it is not entered.</p> <p>Default: None</p> <p>Validation: Should be a valid currency, so that the corresponding currency code can be derived.</p> <p>Error: AR_RAPI_USR_CURR_CODE_INVALID</p>
p_currency_code	IN	VARCHAR2		<p>The actual currency code that gets stored in AR tables.</p> <p>Default:</p> <ol style="list-style-type: none"> 1. Derived from p_usr_currency_code if entered. Otherwise, 2. Defaults to the functional currency code <p>Validation: Validated against the currencies in fnd_currencies table.</p> <p>Error: AR_RAPI_CURR_CODE_INVALID</p> <p>Warning: AR_RAPI_FUNC_CURR_DEFAULTED</p>
p_usr_exchange_rate_type	IN	VARCHAR2		<p>The translated exchange rate type. Used to derive the p_exchange_rate_type if it has not been entered.</p> <p>Default: None</p> <p>Validation: Should be a valid rate type.</p> <p>Error: AR_RAPI_USR_X_RATE_TYP_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_exchange_rate_ty pe	IN	VARCHAR2		<p>Exchange rate type stored in AR tables.</p> <p>Default:</p> <ol style="list-style-type: none"> 1. In case of foreign currency receipt, derived from p_usr_exchange_rate_type. 2. In case of foreign currency receipt, defaults from profile option AR: Default Exchange Rate Type <p>Validation: Validated against values in gl_daily_conversion_types table.</p> <p>Error: AR_RAPI_X_RATE_TYPE_INVALID</p>
p_exchange_rate	IN	NUMBER		<p>The exchange rate between the receipt currency and the functional currency.</p> <p>Default:</p> <ol style="list-style-type: none"> 1. Derived from the Daily Rates table for rate_type <> 'User' in case of non-functional currency 2. If profile option Journals: Display Inverse Rate = 'Y', set user-entered value to 1/p_exchange_rate 3. The entered value is rounded to a precision of 38 <p>Validation:</p> <ol style="list-style-type: none"> 1. In case of non-functional currency, the rate should have a positive value for rate type = 'User' 2. For non-functional currency and type is <> 'User', do not specify any value <p>Error: AR_RAPI_X_RATE_INVALID AR_RAPI_X_RATE_NULL</p>

Parameter	Type	Data-type	Required	Description
p_exchange_rate_date	IN	DATE		<p>The date on which the exchange rate is valid.</p> <p>Default: Receipt date</p> <p>Validation: For a non-functional currency and type is \neq 'User', there should be a valid rate existing in the database for this date. This is a cross validation of type, currency, and date.</p> <p>Error: AR_NO_RATE_DATA_FOUND</p>
p_amount	IN	NUMBER	Yes	<p>The miscellaneous receipt amount.</p> <p>Default: Null</p> <p>Validation: None</p> <p>Error: None</p>
p_receipt_number	IN	VARCHAR2(30)		<p>The receipt number of the receipt to be created.</p> <p>Default: If not specified, the receipt number is defaulted from the document sequence value.</p> <p>Validation: Receipt number should not be null</p> <p>Error: AR_RAPI_RCPT_NUM_NULL</p>
p_receipt_date	IN	DATE		<p>The receipt date of the entered cash receipt.</p> <p>Default: System date</p> <p>Validation: None</p> <p>Error: None</p>

Parameter	Type	Data-type	Required	Description
p_gl_date	IN	DATE		<p>Date when this receipt will be posted to the general ledger.</p> <p>Default: Gets defaulted to the receipt date if it is a valid gl_date, otherwise:</p> <ul style="list-style-type: none"> • If the most recent open period is prior to the receipt date: last date of that period • If there is a period open after the receipt date: first date of the last open period <p>Validation: It is valid if the following conditions are true:</p> <ul style="list-style-type: none"> • The date is in an Open or Future period • The period cannot be an Adjustment period <p>Error: AR_INVALID_APP_GL_DATE</p>

Parameter	Type	Data-type	Required	Description
p_receivables_trx_id	IN	NUMBER(15)		<p>Identifies the receivables activity.</p> <p>Default: If not specified, it is derived from p_activity.</p> <p>Validation: Validates it against the values in the ar_receivables_trx table</p> <ul style="list-style-type: none"> • Type column having values: 'MISCCASH', 'BANK_ERROR', 'CCREFUND'. • Checks the receipt_date to be within start_date_active and end_date_active column values. • Status is Active or null. • Not null. <p>Error: AR_RAPI_ACTIVITY_INVALID AR_RAPI_REC_TRX_ID_INVALID AR_RAPI_REC_TRX_ID_NULL</p>
p_activity	IN	VARCHAR2(50)		<p>Name of the receivables activity. This is used to derive the p_receivables_trx_id.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_misc_payment_so	IN	VARCHAR2(30)		<p>Identifies the source of the miscellaneous receipt.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>

Parameter	Type	Data-type	Required	Description
p_tax_code	IN	VARCHAR2(50)		<p>Depending on the sign of the amount entered, it is the asset tax code (for positive sign or zero) or the liability tax code (negative sign). This is used to derive the p_vat_tax_id.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>

Parameter	Type	Data-type	Required	Description
p_vat_tax_id	IN	NUMBER(15)		<p>The VAT tax identifier for the current miscellaneous receipt.</p> <p>Default:</p> <ul style="list-style-type: none"> • defaulted from p_tax_code • defaulted from receivables_trx_id/activity <p>Validation:</p> <ol style="list-style-type: none"> 1. For 'Accrual' accounting method, the vat_tax_id is validated against the values in ar_vat_tax having <ul style="list-style-type: none"> • receipt_date between start_date_active and end_date_active column values • enabled_flag = 'Y' • tax_type should not be 'TAX_GROUP', 'LOCATION', 'SALES_TAX' • displayed_flag = 'Y' • The tax_class is 'O' (output) for positive or zero amount and 'T' (input) for negative amount • ledger should match the current ledger 2. For 'Cash basis' accounting method, the vat_tax_id should not be specified. <p>Error:</p> <p>AR_RAPI_VAT_TAX_ID_INVALID AR_RAPI_TAX_CODE_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_tax_rate	IN	NUMBER		<p>The new tax rate specified when you override the rate for an ad-hoc tax code.</p> <p>Default:</p> <ol style="list-style-type: none"> 1. Defaulted from the tax rate on the tax code (p_tax_code/p_vat_tax_id). 2. Defaulted from the p_tax_amount when the tax amount is specified for the ad-hoc tax code case. <p>Validation: For 'Accrual' accounting method, tax rate can be specified only in case of an ad-hoc tax code (p_tax_code/p_vat_tax_id) and the profile option 'Tax: Allow Ad Hoc Tax Changes' set to Yes. For 'Cash basis' accounting method, the tax_rate should never be specified.</p> <p>Error: AR_RAPI_TAX_RATE_INVALID AR_RAPI_TAX_RATE_AMT_X_INVALID</p>
p_tax_amount	IN	NUMBER		<p>The tax amount specified in case where you override the rate for an ad-hoc tax code. It is used to derive the tax_rate.</p> <p>Default: None</p> <p>Validation: This needs to be specified only in case of an ad-hoc tax code (p_tax_code/p_vat_tax_id) and the profile option 'Tax: Allow Ad Hoc Tax Changes' set to Yes. For 'Cash basis' accounting method, the tax_amount should never be specified</p> <p>Error: AR_RAPI_TAX_RATE_AMT_X_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_deposit_date	IN	DATE		<p>The deposit date.</p> <p>Default: Receipt date</p> <p>Validation: None</p> <p>Error: None</p>
p_reference_type	IN	VARCHAR2(30)		<p>Indicates whether this miscellaneous receipt is a 'PAYMENT', 'RECEIPT', 'PAYMENT_BATCH' or 'REMITTANCE'.</p> <p>Default: None</p> <p>Validation:</p> <ul style="list-style-type: none"> • Check it for the specified valid values. • Should not have a null value if either p_reference_id or p_reference_num is specified. <p>Error: AR_RAPI_REF_TYPE_INVALID AR_RAPI_REF_TYPE_NULL</p>
p_reference_id	IN	NUMBER(15)		<p>A foreign key to AR_BATCHES, AR_CASH_RECEIPTS, AP_INVOICE_SELECTION_CRITERIA or AP_CHECKS, depending on the specified value of p_reference_type.</p> <p>Default: None</p> <p>Validation: Detailed in Validation, page 8-17.</p> <p>Error: AR_RAPI_REF_NUM_INVALID AR_RAPI_REF_ID_INVALID</p>
p_reference_num	IN	VARCHAR2(30)		<p>The reference number. It is used for deriving the p_reference_id.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>

Parameter	Type	Data-type	Required	Description
p_remittance_bank_account_id	IN	NUMBER(15)		<p>Identifies the user's bank account for depositing the receipt.</p> <p>Default:</p> <ol style="list-style-type: none"> From remittance bank account number From the receipt method based on logic mentioned in Defaulting, page 8-18. <p>Validation: In addition to the validation logic detailed in Validation, page 8-17, those receipt methods which have notes_receivable = 'Y' or bill_of_exchange_flag = 'Y' on the receipt class are excluded for miscellaneous receipts.</p> <p>Error:</p> <p>AR_RAPI_Rem_BK_AC_ID_INVALID AR_RAPI_Rem_BK_AC_ID_NULL</p>
p_remittance_bank_account_num	IN	VARCHAR2(30))	<p>The remittance bank account number. Used to default the remittance bank account id if not specified.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error:</p> <p>AR_RAPI_Rem_BK_AC_NUM_INVALID</p>
p_remittance_bank_account_name	IN	VARCHAR2(50))	<p>The remittance bank account name. Used to default the remittance bank account id if not specified.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error:</p> <p>AR_RAPI_Rem_BK_AC_NAME_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_ussgl_transaction_code	IN	VARCHAR2(30)		<p>Code defined by public sector accounting.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_receipt_method_id	IN	NUMBER(15)		<p>Identifies the receipt method of the receipt.</p> <p>Default: From receipt method name</p> <p>Validation: In addition to the validation logic detailed in Validation, page 8-17, those receipt methods which have notes_receivable = 'Y' or bill_of_exchange_flag = 'Y' on the receipt class are excluded for the miscellaneous receipts.</p> <p>Error: AR_RAPI_INVALID_RCT_MD_ID</p>
p_receipt_method_name	IN	VARCHAR2(30)		<p>The receipt method name of the receipt. Used to default the receipt method id if not specified</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPI_RCPT_MD_NAME_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_doc_sequence_value	IN	NUMBER		<p>Value assigned to document receipt.</p> <p>Default: Detailed in Defaulting, page 8-18.</p> <p>Validation:</p> <ul style="list-style-type: none"> User should not pass in the value if the current document sequence is automatic Document sequence value should not be entered if profile option Sequential Numbering is set to Not Used <p>Error:</p> <p>AR_RAPI_DOC_SEQ_AUTOMATIC AR_RAPI_DOC_SEQ_VAL_INVALID</p>
p_anticipated_clearing_date	IN	DATE		<p>Date the receipt is expected to be cleared.</p> <p>Default: None</p> <p>Validation: greater than or equal to gl_date</p> <p>Error:</p> <p>AR_RW_EFFECTIVE_BEFORE_GL_DATE</p>
p_attribute_rec	IN	attribute_rec_type		<p>This is a record type which contains all 15 descriptive flexfield segments and one descriptive flexfield structure defining column. It represents the Receipt Information flexfield.</p> <p>Default: DFF APIs used to do the defaulting and validation</p> <p>Validation: DFF APIs used to do the defaulting and validation</p> <p>Error: AR_RAPI_DESC_FLEX_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_global_attribute_r_ec	IN	global_attribute_rec_type		<p>This is a record type which contains all 20 global descriptive flexfield segments and one global descriptive flexfield structure defining column.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error:</p>
p_comments	IN	VARCHAR2(240)		User's comments.
p_misc_receipt_id	OUT	NUMBER(15)	Yes	The cash_receipt_id of the receipt created by the API call.
p_called_from	IN	VARCHAR2(20)		<p>This parameter is used to identify the calling routine.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_org_id				

Ar_receipt_api_pub.apply_other_account

Call this routine to do an "other" account application on a cash receipt. Typically this would be to create a claim investigation application with a noninvoice-related deduction or overpayment in Trade Management (if installed).

This API routine has 4 output and 27 input parameters in total. Based on the type, the following is the breakdown of the parameters:

Input

Standard API parameters: 4

Application parameters: 19 + 1 (descriptive flexfield record type)

1 (global descriptive flexfield record type)

Output

Standard API parameters: 3

Application parameters: 1

Parameter Descriptions

For a description of this routine's standard parameters, see
Ar_receipt_api_pub.Create_cash, page 8-3.

The following table lists the descriptions of the other account application-related parameters of the API:

Parameter	Type	Data-type	Required	Description
p_cash_receipt_id	IN	NUMBER (15)		<p>The cash_receipt_id of the receipt which is to be applied to the "other" account.</p> <p>Default: None</p> <p>Validation:</p> <ol style="list-style-type: none">1. Type must be 'CASH'.2. Status must not be Reversed or Approved.3. The receipt must not be Unidentified. <p>Error: AR_RAPI_CASH_RCPT_ID_INVALID AR_RAPI_CASH_RCPT_ID_NULL</p>
p_receipt_number	IN	VARCHAR2 (30)		<p>The receipt number of the receipt to be applied to the "other" account. Used to default the cash_receipt_id.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error : AR_RAPI_RCPT_NUM_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_amount_applied	IN	NUMBER		<p>The amount on the cash receipt that is to be applied to the "other" account.</p> <p>Default: Amount due remaining on the receipt.</p> <p>Validation: Less than or equal to the amount due remaining on the receipt.</p> <p>Error:</p> <p>AR_RAPI_APPLIED_AMT_NULL</p> <p>AR_RW_AMOUNT_LESS_THAN_APP</p>
p_applied_payment_schedule_id	IN	NUMBER (15)	Yes	<p>This payment schedule identifier corresponds to special seeded values, such as -4 (for Claim Investigation).</p> <p>Default:</p> <p>Validation: The value should correspond to the special seeded values, such as -4 (Claim Investigation).</p> <p>Error: AR_RAPI_APP_PS_ID_INVALID</p>
p_receivables_trx_id	IN	NUMBER (15)		<p>Identifier of receivables activity.</p> <p>Default: None</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. Valid database value. 2. The activity_type for the receivables_trx_id should be in sync with the provided applied payment schedule identifier. <p>Error :</p> <p>AR_RAPI_REC_TRX_ID_INVALID</p> <p>AR_RAPI_ACTIVITY_X_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_apply_date	IN	DATE		<p>Date the application was applied.</p> <p>Default:</p> <ol style="list-style-type: none"> 1. Receipt date, if receipt date \geq system date. 2. System date, if receipt date $<$ system date. <p>Validation: apply date \geq receipt date</p> <p>Error: AR_APPLY_BEFORE_RECEIPT</p>
p_apply_gl_date	IN	DATE		<p>Date when this application will be posted to the General Ledger.</p> <p>Default: Defaulted to greater of the receipt date and the system date.</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. Validated as per standard gl date validation described for the gl date in the Create_cash routine. 2. \geq receipt gl date <p>Error:</p> <p>AR_INVALID_APP_GL_DATE</p> <p>AR_RW_GL_DATE_BEFORE_REC_GL</p>
p_ussgl_transaction_code	IN	VARCHAR2 (30)		<p>Code defined by public sector accounting.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>

Parameter	Type	Data-type	Required	Description
p_attribute_rec	IN	attribute_rec_type		<p>This is a record type which contains all the 15 descriptive flexfield segments and one descriptive flexfield structure defining column. It represents the Receipt Application Information flexfield.</p> <p>Default: DFF APIs used to do the defaulting and validation.</p> <p>Validation: DFF APIs used to do the defaulting and validation.</p> <p>Error : AR_RAPI_DESC_FLEX_INVALID</p>
p_global_attribute_rec	IN	global_attribute_rec_type		<p>This is a record type which contains all the global descriptive flexfields: one global descriptive flexfield structure defining column and 20 segments.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_comments	IN	VARCHAR2 (240)		<p>User's comments for the other account application.</p>
p_application_ref_type	IN	VARCHAR2 (30)	Yes	<p>Defines the context of the application reference columns. For Trade Management, the value should be 'CLAIM'.</p> <p>Default: None</p> <p>Validation: Must be 'CLAIM' if a Trade Management deduction is being created (Trade Management must be installed).</p> <p>Error: AR_RAPI_INVALID_APP_REF</p>
p_application_ref_id	IN	NUMBER (15)		<p>Not used. Leave null.</p>

Parameter	Type	Data-type	Required	Description
p_application_ref_n um	IN	VARCHAR2 (30)		<p>The reference number relating to the application reference type. If application reference type is 'CLAIM', then this would be a deduction number.</p> <p>Default: None</p> <p>Validation: If populated, then must be an existing deduction number in Trade Management.</p> <p>Error: AR_RAPI_INVALID_CLAIM_NUM</p>
p_secondary_applica tion_ref_id	IN	NUMBER (15)		<p>The secondary application reference ID related to the application reference type.</p> <p>Default: None</p> <p>Validation: If populated, and if application reference type is 'CLAIM', then this must contain a valid claim ID in Trade Management.</p> <p>Error: AR_RW_INVALID_CLAIM_ID</p>
p_payment_set_id	IN	NUMBER (15)		<p>Payment set ID is populated only for a prepayment receipt which is to be applied to the "other" account.</p> <p>Default: None</p> <p>Validation: None</p>
p_receivable_applica tion_id	OUT	NUMBER (15)		<p>The ID of the resulting activity receivable application.</p>
p_application_ref_re ason	IN	VARCHAR2 (30)		<p>The reason code related to the application reference type.</p> <p>Default: None</p> <p>Validation: If populated, and if application reference type is 'CLAIM', then this must contain a valid reason code ID from Trade Management.</p> <p>Error: AR_RAPI_INVALID_REF_REASON</p>
p_customer_referenc e	IN	VARCHAR2 (100)		<p>Customer supplied reference.</p>

Parameter	Type	Data-type	Required	Description
p_customer_reason	IN	VARCHAR2 (30)		<p>Reason code supplied by customer, in the context of an application reference type of 'CLAIM'.</p> <p>Default: None</p> <p>Validation: None in Oracle Receivables (the attempt to match to an Oracle reason code is made in Trade Management).</p>
p_called_from	IN	VARCHAR2 (20)		<p>This parameter is used to identify the calling routine.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_org_id				

Example

Objective:

To apply a cash receipt in functional currency to Claim Investigation, and to create a non-invoice overpayment in the functional currency using a call to the API *Ar_receipt_api_pub.apply_other_account* and passing a minimum number of input parameters.

This table lists the entered parameters:

Parameter	Entered Value	Default Value
p_api_version	1.0	
p_receipt_number	'aj_test_cr_2'	
p_receivables_trx_id	1400	
p_application_ref_type	'CLAIM'	
p_applied_payment_schedule_id	-4	

This table lists the defaulted input parameters, which were not entered:

Parameter	Entered Value	Default Value
p_cash_receipt_id		23927
p_gl_date		01-JUN-2000
p_apply_date		01-JUN-2000
p_amount_applied		100

The retrieval and handling of the warnings and error messages, put on the message stack by the API during execution, are the same as described in Defaulting, page 8-18.

Ar_receipt_api_pub.unapply_other_account

Call this routine to do a reversal of an "other" account application on a cash receipt.

This API routine has 3 output and 11 input parameters in total. Based on the type, the following is the breakdown of the parameters:

Input

Standard API parameters: 4

Application parameters: 7

Output

Standard API parameters: 3

Application parameters: 0

Parameter Descriptions

For a description of this routine's standard parameters, see
Ar_receipt_api_pub.Create_cash, page 8-3.

The following table lists the descriptions of the other account unapplication-related parameters of the API:

Parameter	Type	Data-type	Required	Description
p_cash_receipt_id	IN	NUMBER (15)		<p>The cash_receipt_id of the receipt which is to be applied to the "other" account.</p> <p>Default: None</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. Type must be 'CASH'. 2. Status must not be Reversed or Approved. 3. The receipt must not be Unidentified. <p>Error:</p> <p>AR_RAPI_CASH_RCPT_ID_INVALID</p> <p>AR_RAPI_CASH_RCPT_ID_NULL</p>
p_receipt_number	IN	VARCHAR2 (30)		<p>The receipt number of the receipt from which the "other" account application is to be unapplied. Used to default the cash_receipt_id.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPI_RCPT_NUM_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_receivable_application_id	IN	NUMBER (15)		<p>Identifies the receivable application. Used to derive the customer trx id, cash_receipt_id, and the applied_ps_id, if not specified.</p> <p>Default: Refer to Validation, page 8-73.</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. Application type = 'CASH'. 2. Display flag = 'Y' (latest application) and status = 'OTHER ACC'. 3. The applied payment schedule id of the receivable application record must correspond to the p_applied_payment_schedules_id, if specified. 4. The cash receipt id must correspond to the cash receipt id specified. <p>Error: AR_RAPI_REC_APP_ID_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_reversal_gl_date	IN	DATE		<p>The reversal gl date.</p> <p>Default: Gets defaulted to the application gl date if it is a valid gl_date.</p> <p>Validation:</p> <p>It is valid if the following conditions are true:</p> <ul style="list-style-type: none"> • The date is in an Open or Future period. • The period cannot be an Adjustment period. • The reversal GL date \geq application GL date. • The reversal GL date \geq receipt GL date. <p>If the date is invalid, then:</p> <ul style="list-style-type: none"> • If the most recent open period is prior to the receipt date: last date of that period • If there is a period open after the receipt date: first date of the last open period <p>Error: AR_INVALID_APP_GL_DATE AR_RW_BEFORE_APP_GL_DATE AR_RW_BEFORE_RECEIPT_GL_DATE</p>
p_called_from	IN	VARCHAR2 (20)		<p>Indicates which program is calling this API.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_cancel_claim_flag	IN	VARCHAR2 (1)		Not used. Leave null.
p_org_id				

Example

Objective:

To unapply an "other" account application using the call to API *Ar_receipt_api_pub.unapply_other_account* and passing a minimum number of input parameters.

This table lists the entered parameters:

Parameter	Entered Value	Default Value
p_api_version	1.0	
p_receivable_application_id	10053	

This table lists the defaulted input parameters, which were not entered:

Parameter	Entered Value	Default Value
p_cash_receipt_id		20338
p_reversal_gl_date		01-JUN-2000

The retrieval and handling of the warnings and error messages, put on the message stack by the API during execution, are the same as described in Defaulting, page 8-18.

Ar_receipt_api_pub.apply_open_receipt

Call this routine to apply a cash receipt to another open receipt. Open receipts include unapplied cash, on-account cash, and claim investigation applications. Claim investigation applications can be applied only if Trade Management is installed.

This API routine has 8 output and 19 input parameters in total. Based on the type, the following is the breakdown of the parameters:

Input

Standard API parameters: 4

Application parameters: 13 + 2 (descriptive and global descriptive flexfield record type)

Output

Standard API parameters: 3

Application parameters: 5

Parameter Descriptions

For a description of this routine's standard parameters, see `Ar_receipt_api_pub.Create_cash`, page 8-3.

The following table lists the descriptions of the apply open receipt-related parameters of the API:

Parameter	Type	Data-type	Required	Description
<code>p_cash_receipt_id</code>	IN	NUMBER (15)		<p>The <code>cash_receipt_id</code> of the receipt which is to be applied to an open receipt.</p> <p>Default: None</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. Type must be 'CASH'. 2. Status must not be Reversed or Approved. 3. The receipt must not be Unidentified. 4. The receipt being applied and the open receipt must have the same currency. <p>Error:</p> <p><code>AR_RAPI_CASH_RCPT_ID_INVALID</code> <code>AR_RAPI_CASH_RCPT_ID_NULL</code> <code>AR_RW_NET_DIFF_RCT_CURR</code></p>
<code>p_receipt_number</code>	IN	VARCHAR2 (30)		<p>The receipt number of the receipt to be applied to an open receipt. Used to default the <code>cash_receipt_id</code>. The receipt being applied and the open receipt must have the same currency.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: <code>AR_RAPI_RCPT_NUM_INVALID</code> <code>AR_RW_NET_DIFF_RCT_CURR</code></p>
<code>p_applied_payment_schedule_id</code>	IN	NUMBER (15)		Not used. Leave null.

Parameter	Type	Data-type	Required	Description
p_open_cash_receipt_id	IN	NUMBER (15)		<p>The cash_receipt_id of the open receipt which is to be applied to.</p> <p>Default: None</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. Type must be 'CASH'. 2. Status must not be Reversed or Approved. 3. The receipt must not be Unidentified. 4. The receipt being applied and the open receipt must have the same currency. <p>Error:</p> <p>AR_RAPI_CASH_RCPT_ID_INVALID AR_RAPI_CASH_RCPT_ID_NULL AR_RW_NET_DIFF_RCT_CURR</p>
p_open_receipt_num	IN	VARCHAR2 (30)		<p>The receipt number of the open receipt. Used to default the open cash_receipt_id.</p> <p>The receipt being applied and the open receipt must have the same currency.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPI_RCPT_NUM_INVALID AR_RW_NET_DIFF_RCT_CURR</p>
p_open_rec_app_id	IN	NUMBER (15)		<p>The ID of the receivable application of the open receipt, if on-account or claim investigation.</p> <p>Default: None</p> <p>Validation: Must have status of ACC or OTHER ACC, and display must be 'Y'.</p> <p>Errors: AR_RAPI_REC_APP_ID_INVALID AR_RW_NET_OPEN_RCT_ONLY</p>

Parameter	Type	Data-type	Required	Description
p_amount_applied	IN	NUMBER (15)		<p>The amount on the cash receipt that is to be applied to an open receipt.</p> <p>Default: None</p> <p>Validation: Must be a natural application, i.e. it must move the balance on the open receipt closer to zero.</p> <p>Error:</p> <p>AR_RAPI_APPLIED_AMT_NULL</p> <p>AR_RW_AMOUNT_LESS_THAN_APP</p> <p>AR_RW_NET_OPEN_AMT_INC</p>
p_apply_date	IN	DATE		<p>Date the application was applied.</p> <p>Default:</p> <ol style="list-style-type: none"> 1. Receipt date, if receipt date \geq system date. 2. System date, if receipt date $<$ system date. <p>Validation: apply date \geq receipt date.</p> <p>Error: AR_APPLY_BEFORE_RECEIPT</p>

Parameter	Type	Data-type	Required	Description
p_apply_gl_date	IN	DATE		<p>Date when this application will be posted to the General Ledger.</p> <p>Default: Defaulted to greater of the receipt GL date, the open receipt GL date, and the system date.</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. Validated as per standard gl date validation described for the gl date in the Create_cash routine. 2. \geq receipt gl date. <p>Error:</p> <p>AR_INVALID_APP_GL_DATE</p> <p>AR_RW_GL_DATE_BEFORE_REC_GL</p> <p>AR_RW_GL_DATE_BEFORE_OPEN_REC</p>
p_ussgl_transaction_code	IN	VARCHAR2 (30)		<p>Code defined by public sector accounting.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_attribute_rec	IN	attribute_rec_type		<p>This is a record type which contains all the 15 descriptive flexfield segments and one descriptive flexfield structure defining column. It represents the Receipt Application Information flexfield.</p> <p>Default: DFF APIs used to do the defaulting and validation.</p> <p>Validation: DFF APIs used to do the defaulting and validation.</p> <p>Error: AR_RAPI_DESC_FLEX_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_global_attribute_rec	IN	global_attribute_rec_type		<p>This is a record type which contains all the global descriptive flexfields: One global descriptive flexfield structure defining column and 20 segments.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_comments	IN	VARCHAR2 (240)		User's comments for the other account application.
x_application_ref_num	OUT	VARCHAR2 (30)		The reference number from the open receipt application, if applicable. If the application reference type is 'CLAIM', then this would be a deduction number.
x_receivable_application_id	OUT	NUMBER (15)		The ID of the resulting payment netting receivable application.
x_applied_rec_app_id	OUT	NUMBER (15)		The ID of the corresponding payment netting receivable application created on the applied-to receipt.
x_acctd_amount_applied_from	OUT	NUMBER (15)		Amount applied from the receipt, in functional currency and converted using the main receipt's exchange rate.
x_acctd_amount_applied_to	OUT	VARCHAR2 (30)		Amount applied to the open receipt, in functional currency and converted using the open receipt's exchange rate. Used in conjunction with x_applied_amount_applied_from to determine exchange gain/loss.
p_called_from	IN	VARCHAR2 (20)		<p>This parameter is used to identify the calling routine.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>

Parameter	Type	Data-type	Required	Description
p_org_id				

Example

Objective:

To apply a cash receipt in your functional currency to unapplied cash on another receipt, using a call to the API *Ar_receipt_api_pub.apply_open_receipt* and passing a minimum number of input parameters.

This table lists the entered parameters:

Parameter	Entered Value	Default Value
p_api_version	1.0	
p_receipt_number	'aj_test_cr_10'	
p_open_receipt_number	'aj_test_cr_30'	
p_amount_applied	-200	

This table lists the defaulted input parameters, which were not entered:

Parameter	Entered Value	Default Value
p_cash_receipt_id		23935
p_open_cash_receipt_id		23973
p_gl_date		01-JUN-2000
p_apply_date		01-JUN-2000

The retrieval and handling of the warnings and error messages, put on the message stack by the API during execution, are the same as described in Defaulting, page 8-18.

Ar_receipt_api_pub.unapply_open_receipt

Call this routine to reverse a payment netting application on a cash receipt.

This API routine has 3 output and 8 input parameters in total. Based on the type, the following is the breakdown of the parameters:

Input

Standard API parameters: 4

Application parameters: 4

Output

Standard API parameters: 3

Application parameters: 0

Parameter Descriptions

For a description of this routine's standard parameters, see [Ar_receipt_api_pub.Create_cash](#), page 8-3.

The following table lists the descriptions of the unapply open receipt-related parameters of the API:

Parameter	Type	Data-type	Required	Description
p_receivable_application_id	IN	NUMBER(15)		<p>Identifies the receivable application to be unapplied.</p> <p>Default: Refer to Validation, page 8-73.</p> <p>Validation:</p> <ol style="list-style-type: none">1. Application type = 'CASH'.2. Display flag = 'Y' (latest application) and status = 'ACTIVITY', receivables_trx_id = -163.3. Unapplying this application must not result in either receipt becoming negative. <p>Error: AR_RAPI_REC_APP_ID_INVALID AR_RW_NET_UNAPP_OVERAPP</p>

Parameter	Type	Data-type	Required	Description
p_reversal_gl_date	IN	DATE		<p>The reversal gl date.</p> <p>Default: Gets defaulted to the application gl date if it is a valid gl_date.</p> <p>Validation:</p> <p>It is valid if the following conditions are true:</p> <ul style="list-style-type: none"> • The date is in an Open or Future period. • The period cannot be an Adjustment period. • The reversal GL date \geq application GL date. • The reversal GL date \geq receipt GL date. <p>If the date is invalid, then:</p> <ul style="list-style-type: none"> • If the most recent open period is prior to the receipt date: last date of that period • If there is a period open after the receipt date: first date of the last open period <p>Error: AR_INVALID_APP_GL_DATE AR_RW_BEFORE_APP_GL_DATE AR_RW_BEFORE_RECEIPT_GL_DATE</p>
p_called_from	IN	VARCHAR2(20)	Yes	<p>Indicates which program is calling this API.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_org_id				

Example

Objective:

To unapply an open receipt/payment netting application using the call to API *Ar_receipt_api_pub.unapply_open_receipt* and passing a minimum number of input

parameters.

This table lists the entered parameters:

Parameter	Entered Value	Default Value
p_api_version	1.0	
p_receivable_application_id	10055	

This table lists the defaulted input parameters, which were not entered:

Parameter	Entered Value	Default Value
p_reversal_gl_date		01-JUN-2000

The retrieval and handling of the warnings and error messages, put on the message stack by the API during execution, are the same as described in Defaulting, page 8-18.

Ar_receipt_api_pub.Create_apply_on_acc

This routine is called to create a cash receipt and place it on account. Use this routine when no specific debit item is referenced for receipt application, but you do not want to leave the cash as an unapplied liability.

This is essentially a superset of Ar_receipt_api_pub.Create_cash, page 8-3 and Ar_receipt_api_pub.Apply_on_account, page 8-60 APIs, and contains the same parameters as contained in those two APIs. During the call to this API, if the receipt is successfully created but its on-account application fails, then the receipt creation is also rolled back.

This routine calls Oracle Payments, where required. See Integration with Oracle Payments, page 8-2.

Note: To create credit card receipts that need to be processed by Oracle Payments APIs, you must pass the p_call_payment_processor parameter as fnd_api.g_true. Additionally, you must specify the p_customer_bank_account_id parameter.

This API routine has 4 output and 57 input parameters:

Input

Standard API parameters: 4

Application parameters: 49 + 2 (descriptive flexfield parameter)
+ 2 (global descriptive flexfield parameter)

Output

Standard API parameters: 3

Application parameters: 1

Parameter Descriptions

For a description of this routine's standard parameters, see
`Ar_receipt_api_pub.Create_cash`, page 8-3.

The following table lists the parameters that pertain specifically to the receipt creation and on-account application routine:

Parameter	Type	Data-type	Required	Description
<code>p_usr_currency_code</code>	IN	VARCHAR2		<p>The translated currency code.</p> <p>Used to derive the <code>p_currency_code</code> if it is not entered.</p> <p>Default: None</p> <p>Validation: Should be a valid currency, so that the corresponding currency code can be derived.</p> <p>Error: <code>AR_RAPI_USR_CURR_CODE_INVALID</code></p>
<code>p_currency_code</code>	IN	VARCHAR2 (15)		<p>The actual currency code that gets stored in AR tables.</p> <p>Default:</p> <ol style="list-style-type: none">1. Derived from <code>p_usr_currency_code</code> if entered, else2. Defaults to the functional currency code <p>Validation: Validated against the currencies in the <code>fnd_currencies</code> table.</p> <p>Error: <code>AR_RAPI_CURR_CODE_INVALID</code></p> <p>Warning: <code>AR_RAPI_FUNC_CURR_DEFAULTED</code></p>

Parameter	Type	Data-type	Required	Description
p_usr_exchange_rate_type	IN	VARCHAR2		<p>The translated exchange rate type.</p> <p>Used to derive the p_exchange_rate_type if it has not been entered.</p> <p>Default: None</p> <p>Validation: Should be a valid rate type.</p> <p>Error: AR_RAPI_USR_X_RATE_TYP_INVALID</p>
p_exchange_rate_ty_pe	IN	VARCHAR2 (30)		<p>Exchange rate type stored in AR tables.</p> <p>Default:</p> <ol style="list-style-type: none"> 1. In case of foreign currency receipt, derived from p_usr_exchange_rate_type. 2. In case of foreign currency receipt, defaults from AR: Default Exchange Rate Type profile option. 3. Should be left null, if receipt is in the same denomination as functional currency. <p>Validation: Validated against values in gl_daily_conversion_types table</p> <p>Error: AR_RAPI_X_RATE_TYPE_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_exchange_rate	IN	NUMBER		<p>The exchange rate between the receipt currency and the functional currency.</p> <p>Default:</p> <ol style="list-style-type: none"> 1. Derived from the Daily Rates table for rate_type \diamond 'User' in case of non-functional currency. 2. If profile option Journals: Display Inverse Rate = 'Y', set user-entered value to 1/ p_exchange_rate. 3. The entered value is rounded to a precision of 38. <p>Validation:</p> <ol style="list-style-type: none"> 1. In case of non-functional currency, the rate should have a positive value for rate type='User'. 2. For non-functional currency and type \diamond 'User', do not specify any value. <p>Error: AR_RAPI_X_RATE_INVALID AR_RAPI_X_RATE_NULL</p>
p_exchange_rate_date	IN	DATE		<p>The date on which the exchange rate is valid.</p> <p>Default: Receipt date</p> <p>Validation: For a non-functional currency and type \diamond 'User', a valid rate should exist in the database for this date. This is a cross validation of type, currency, and date.</p> <p>Error: AR_NO_RATE_DATA_FOUND</p>
p_amount	IN	NUMBER	Yes	<p>The cash receipt amount.</p> <p>Default: Null</p> <p>Validation: > 0</p> <p>Error: AR_RAPI_REC_AMT_NEGATIVE AR_RAPI_RCPT_AMOUNT_NULL</p>

Parameter	Type	Data-type	Required	Description
p_factor_discount_amount	IN	NUMBER		<p>The bank charges on the cash receipt. Default: None</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. Bank charges are not allowed if profile option AR: Create Bank Charges = 'No'. 2. Bank charges not allowed if the receipt state, derived from the receipt class of the receipt method, \diamond 'Cleared'. 3. If allowed, then ≥ 0. <p>Error: AR_BK_CH_NOT_ALLWD_IF_NOT_CLR AR_JG_BC_AMOUNT_NEGATIVE</p>
p_receipt_number	IN	VARCHAR2(30)		<p>The receipt number of the receipt to be created. Default: If not specified, the receipt number is defaulted from the document sequence value. Validation: Receipt number should not be null. Error: AR_RAPI_RCPT_NUM_NULL</p>
p_receipt_date	IN	DATE		<p>The receipt date of the entered cash receipt. Default: System date Validation: None Error: None</p>

Parameter	Type	Data-type	Required	Description
p_gl_date	IN	DATE		<p>Date that this receipt will be posted to the general ledger.</p> <p>Default: Gets defaulted to the receipt date if it is a valid gl_date.</p> <p>Validation: The date is valid if the following conditions are true:</p> <ul style="list-style-type: none"> • The date is in an Open or Future period • The period cannot be an Adjustment period <p>If the date is invalid, then:</p> <ul style="list-style-type: none"> • If the most recent open period is prior to the receipt date: last date of that period • If there is a period open after the receipt date: first date of the last open period <p>Error: AR_INVALID_APP_GL_DATE</p>
p_maturity_date	IN	DATE		<p>Receipt maturity date.</p> <p>Default: Deposit date</p> <p>Validation: \geq p_receipt_date</p> <p>Error: AR_RW_MAT_BEFORE_RCT_DATE</p>
p_customer_id	IN	NUMBER (15)		<p>The customer_id for the paying customer.</p> <p>Default: Refer to Defaulting, page 8-52.</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. Customer exists and has prospect code = 'CUSTOMER' 2. Customer has a profile defined at the customer level <p>Error: AR_RAPI_CUST_ID_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_customer_name	IN	VARCHAR2 (50)		<p>The name for the entered customer. Used to default the customer id if not specified.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPI_CUS_NAME_INVALID</p>
p_customer_number	IN	VARCHAR2 (30)		<p>The customer number. Used to default the customer_id if not specified.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPI_CUS_NUM_INVALID</p>
p_customer_bank_ac count_id	IN	NUMBER (15)		<p>The customer bank account id.</p> <p>Default: From bank account id/number</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. It must be a valid bank account of the paying customer . 2. The inactive date (if defined) of the bank account should be greater than the receipt_date. 3. The receipt date must be within the Start date and the End date of the bank account uses. <p>Error: AR_RAPI_CUS_BK_AC_2_INVALID AR_RAPI_CUS_BK_AC_ID_INVALID</p>
p_customer_bank_ac count_num	IN	VARCHAR2 (30)		<p>The customer bank account number. Used to default the customer bank account id, if not specified.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>

Parameter	Type	Data-type	Required	Description
p_customer_bank_ac count_name	IN	VARCHAR2 (80)		<p>The customer bank account name. Used to default the customer bank account id, if not specified.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_location	IN	VARCHAR2 (40)		<p>The Bill_To location for the customer. Used to derive the p_customer_site_use_id.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPI_CUS_LOC_INVALID</p>
p_customer_site_use _id	IN	NUMBER (15)		<p>The Bill_To site_use_id for the customer.</p> <p>Default:</p> <ol style="list-style-type: none"> 1. Defaulted from customer location. Otherwise, <ol style="list-style-type: none"> 2. Primary Bill_To customer site_use_id of the customer. <p>Validation: It should be a valid Bill_To site of the paying customer.</p> <p>Error: AR_RAPI_CUS_SITE_USE_ID_INVALID</p>
p_customer_receipt_ reference	IN	VARCHAR2 (30)		<p>This column is used to store a customer receipt reference value supplied by the customer at the confirmation time.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>

Parameter	Type	Data-type	Required	Description
p_override_remit_ba_nk_account_flag	IN	VARCHAR2 (1)		<p>The flag value decides when the remittance bank account is can be overridden by the remittance selection process.</p> <p>Default: 'Y'</p> <p>Validation: valid values 'Y' and 'N'</p> <p>Error: AR_RAPI_INVALID_OR_REMIT_BK_AC</p>
p_remittance_bank_account_id	IN	NUMBER (15)		<p>Identifies the user's bank account for depositing the receipt.</p> <p>Default:</p> <ol style="list-style-type: none"> From remittance bank account number From the receipt method based on logic mentioned in Defaulting, page 8-18. <p>Validation: Validation logic detailed in Validation, page 8-17.</p> <p>Error: AR_RAPIREM_BK_AC_ID_INVALID AR_RAPIREM_BK_AC_ID_NULL</p>
p_remittance_bank_account_num	IN	VARCHAR2 (30)		<p>The remittance bank account number. Used to default the remittance bank account id, if not specified.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPIREM_BK_AC_NUM_INVALID</p>
p_remittance_bank_account_name	IN	VARCHAR2 (50)		<p>The remittance bank account name. Used to default the remittance bank account id, if not specified.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: AR_RAPIREM_BK_AC_NAME_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_deposit_date	IN	DATE		<p>The deposit date.</p> <p>Default: receipt date</p> <p>Validation: None</p> <p>Error: None</p>
p_receipt_method_id	IN	NUMBER (15)		<p>Identifies the receipt method of the receipt.</p> <p>Default: From receipt method name.</p> <p>Validation: Validation detailed in Validation, page 8-17.</p> <p>Error: AR_RAPI_INVALID_RCT_MD_ID</p>
p_receipt_method_name	IN	VARCHAR2 (30)		<p>The receipt method name of the receipt. Used to default the receipt method id if not specified.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p> <p>Note: To use credit card refund functionality, ensure that remittance of the original receipt is performed within Oracle Receivables. Do this by setting the remittance method on the receipt method's associated receipt class to <i>Standard</i>.</p> <p>Warning: If you use this API to both authorize and capture credit card payments, then set the remittance method to <i>None</i>. Note, however, that with this setting, you cannot use standard credit card refund functionality. Instead, you must refund such payments <i>outside</i> Receivables.</p>

Parameter	Type	Data-type	Required	Description
p_doc_sequence_value	IN	NUMBER		<p>Value assigned to document receipt.</p> <p>Default: Detailed in Defaulting, page 8-18.</p> <p>Validation:</p> <ul style="list-style-type: none"> User should not pass in the value if the current document sequence is automatic. Document sequence value should not be entered if profile option Sequential Numbering is set to Not Used. <p>Error: AR_RAPI_DOC_SEQ_AUTOMATIC AR_RAPI_DOC_SEQ_VAL_INVALID</p>
p_ussgl_transaction_code	IN	VARCHAR2 (30)		<p>Code defined by public sector accounting.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_anticipated_clearing_date	IN	DATE		<p>Date the receipt is expected to be cleared.</p> <p>Default: None</p> <p>Validation: \geq gl_date</p> <p>Error: AR_RW_EFFECTIVE_BEFORE_GL_DATE</p>
p_event	IN	VARCHAR2		<p>The event that resulted in the creation of the receipt. Currently used only by Bills Receivable.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>

Parameter	Type	Data-type	Required	Description
p_called_from	IN	VARCHAR2 (20)		<p>This parameter is used to identify the calling routine. Currently used to identify only the 'BR_REMIT' program.</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_attribute_record	IN	attribute_rec_type		<p>This is a record type which contains all the 15 descriptive flexfield segments and one descriptive flexfield structure defining column. It represents the Receipt Information flexfield.</p> <p>Default: DFF APIs complete the defaulting and validation.</p> <p>Validation: DFF APIs complete the defaulting and validation.</p> <p>Error: AR_RAPI_DESC_FLEX_INVALID</p>
p_global_attribute_record	IN	global_attribute_rec_type		<p>This is a record type which contains all the 20 global descriptive flexfield segments and one global descriptive flexfield structure defining column.</p> <p>Default: None</p> <p>Validation: None</p>
p_receipt_comments	IN	VARCHAR2 (240)		User's comments for the application.
p_issuer_name	IN	VARCHAR2 (50)		<p>Issuer name of notes receivable (Asia Pacific requirement).</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>

Parameter	Type	Data-type	Required	Description
p_issue_date	IN	DATE		<p>Date when notes receivable was issued (Asia Pacific requirement).</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_issuer_bank_branch_id	IN	NUMBER (15)		<p>Bank/ Branch issuing the notes receivable (Asia Pacific Requirement).</p> <p>Default: None</p> <p>Validation: None</p> <p>Error: None</p>
p_cr_id	OUT	NUMBER (15)		<p>The cash_receipt_id of the receipt created by the API call.</p>
p_amount_applied	IN	NUMBER		<p>The amount on the cash receipt that is to be applied to an account.</p> <p>Default: The default amount applied can be either the open amount of the transaction or the unapplied amount of the receipt, but you can change it. Use the AR: Always Default Transaction Balance for Applications profile option, <i>Oracle Receivables Implementation Guide</i> to control how Receivables defaults the applied amount.</p> <p>For the profile option's defaulting rules, see Ar_receipt_api_pub.Apply, page 8-20.</p> <p>Validation: Less than or equal to the amount due remaining on the receipt.</p> <p>Error: AR_RAPI_APPLIED_AMT_NULL AR_RW_AMOUNT_LESS_THAN_APP</p>

Parameter	Type	Data-type	Required	Description
p_apply_date	IN	DATE		<p>Date the application was applied.</p> <p>Default:</p> <ol style="list-style-type: none"> 1. Receipt date, if receipt date \geq system date. 2. System date, if receipt date $<$ system date. <p>Validation: apply date \geq receipt date</p> <p>Error: AR_APPLY_BEFORE_RECEIPT</p>
p_apply_gl_date	IN	DATE		<p>Date that this application will be posted to the general ledger.</p> <p>Default: Defaulted to greater of the receipt date and the system date.</p> <p>Validation:</p> <ol style="list-style-type: none"> 1. Validated as per standard gl date validation described for the gl date in create_cash routine 2. \geq receipt gl date <p>Error: AR_INVALID_APP_GL_DATE AR_RW_GL_DATE_BEFORE_REC_GL</p>
p_app_ussgl_transac tion_code	IN	VARCHAR2 (30)		<p>Code defined by public sector accounting.</p> <p>Default: None</p> <p>Validation: None</p>
p_app_attribute_rec ord	IN	attribute_rec_t ype		<p>This is a record type which contains all the 15 descriptive flexfield segments and one descriptive flexfield structure defining column. It represents the Receipt Application Information flexfield.</p> <p>Default: DFF APIs complete the defaulting and validation.</p> <p>Validation: DFF APIs complete the defaulting and validation.</p> <p>Error: AR_RAPI_DESC_FLEX_INVALID</p>

Parameter	Type	Data-type	Required	Description
p_app_global_attribute_record	IN	global_attributes_rec_type		<p>This is a record type which contains all the 20 global descriptive flexfield segments and one global descriptive flexfield structure defining column.</p> <p>Default: None</p> <p>Validation: None</p>
app_comments	IN	VARCHAR2 (240)		User's comments for the application.
p_application_ref_number	IN	VARCHAR2 (30)		Deduction number, if resulting from Trade Management claim settlement.
p_secondary_application_ref_id	IN	NUMBER (15)		Claim ID, if resulting from Trade Management claim settlement.
p_customer_reference	IN	VARCHAR2 (100)		Reference supplied by customer.
p_customer_reason	IN	VARCHAR2 (20)		Reason code supplied by customer.
p_secondary_app_receipt_type	IN	VARCHAR2 (30)		Used for automated receipt handling, leave null.
p_secondary_app_receipt_num	IN	VARCHAR2 (30)		Used for automated receipt handling, leave null.
p_call_payment_processor	IN	VARCHAR2 (1)		<p>This is the payment processing indicator flag. Pass as FND_API.G_TRUE, if you want to call i Payment payment APIs for credit card processing.</p>
p_default_site_use	IN	VARCHAR2	No	<p>Indicates if you want to default the site use from p_customer_site_use_id.</p> <p>The default value is Y. Pass N to default nothing.</p> <p>If the Require Billing Location for Receipts system option is selected, then no value is required here.</p>

Example

Objective:

To create a cash receipt and apply to On Account in the functional currency using a call to the API Ar_receipt_api_pub.Create_Apply_on_acc and passing a minimum number of input parameters.

This table lists the entered parameters:

Parameter	Entered Value	Default Value
p_api_version	1.0	
p_receipt_number	'aj_test_api_3'	
p_amount	1000	
p_receipt_method_id	1001	
p_customer_name	'Computer Service and Rentals'	

This table lists the defaulted input parameters, which were not entered:

Parameter	Entered Value	Default Value
p_customer_id		1006
p_currency_code		USD
p_receipt_date		19-APR-2004
p_gl_date		19-APR-2004
p_deposit_date		19-APR-2004
p_customer_site_use_id		1025
p_override_remit_bank_acco unt_flag		'Y'

Parameter	Entered Value	Default Value
p_remittance_bank_account_id	10001	
p_maturity_date	19-APR-2004	
p_apply_gl_date	19-APR-2004	
p_apply_date	19-APR-2004	
p_amount_applied	1000	
p_amount_applied_from	1000	
p_call_payment_processor*	fnd_api.g_false	

Result:

We were able to create the cash receipt 'aj_test_api_3' and then apply it to 'On account' by specifying only 5 input parameters in our call to this API. The receipt is in the functional currency. The retrieval and handling of the warnings and the error messages, put on the message stack by the API during execution, are the same as described in Defaulting, page 8-18.

Messages

Messages play an important role in the effectiveness of your API calls. The right message is raised at the right point to convey to you the exact error that has occurred or any warnings that have been raised.

The Receipt API puts on the message stack all error messages and warnings raised during execution. You can retrieve messages and warnings as described in Exception Handling and Result Messages, page 1-3.

WARNINGS AND ERRORS

The following table lists all the error messages raised by the Receipt API:

TYPE

E: Error message

W: Warning message

MESSAGE_CODE	MESSAGE_TEXT	EXPLANATION	TYPE
AR_APPLY_BEFORE_RECEIPT	Apply Date must be greater than or equal to the Receipt Date.		E
AR_APPLY_BEFORE_TRANSACTION	Apply Date must be greater than or equal to the Transaction Date.		E
AR_BK_CH_NOT_ALLWD_IF_NOT_CLR	For a receipt status other than cleared, bank charges are not allowed.		E
AR_EXCHANGE_RATE_NEGATIVE	Please enter a positive exchange rate.		E
AR_EXCHANGE_RATE_ZERO	The exchange rate cannot be zero.		E
AR_INVALID_APP_GL_DATE	GL date, &GL_DATE, is not in an open or future-enterable period.		E
AR_JG_BC_AMOUNT_NEGATIVE	The Bank Charges amount cannot be negative.		E
AR_NO_PARTIAL_DISC	No discounts allowed on this installment unless it is fully paid.		E
AR_NO_RATE_DATA_FOUND	There is no rate for this currency, rate date and rate type in the database.		E
AR_OVERR_REM_BK_FLAG_INVALID	Override remittance bank flag has invalid value.		E
AR_RAPI_CUS_BK_NAME_NUM_INVALID	Customer bank account identifier has taken a precedence over the customer bank account name and number.		W
AR_RAPI_ACTIVITY_INVALID	The receivables activity name is invalid.		E

MESSAGE_CODE	MESSAGE_TEXT	EXPLANATION	TYPE
AR_RAPI_ACTIVITY_IGN	Both a receivables transaction identifier and a receivables activity exist for this record. The receivables transaction identifier takes precedence over the receivables activity.		W
AR_RAPI_TAX_RATE_AMT_X_IN VALID	Please enter a different combination of receipt amount, tax amount, and tax rate.		E
AR_RAPI_TAX_CODE_INVALID	The tax code is invalid.		E
AR_RAPI_TAX_RATE_INVALID	The tax rate is invalid.		E
AR_RAPI_TAX_CODE_IGN	Both a VAT identifier and a tax code exist for this record. The VAT identifier takes precedence over the tax code.		W
AR_RAPI_REC_TRX_ID_NULL	Please enter a receivables transaction identifier.		E
AR_RAPI_VAT_TAX_ID_INVALID	The VAT identifier is invalid.		E
AR_RAPI_REF_TYPE_INVALID	The reference type is invalid.		E
AR_RAPI_REF_NUM_INVALID	The reference number is invalid.		E
AR_RAPI_REF_NUM_IGN	Both a reference identifier and a reference number exist for this record. The reference identifier takes precedence over the reference number.		W
AR_RAPI_REF_ID_INVALID	The reference identifier is invalid.		E
AR_RAPI_REF_ID_NULL	Please enter a reference identifier.		E

MESSAGE_CODE	MESSAGE_TEXT	EXPLANATION	TYPE
AR_RAPI_REF_TYPE_NULL	Please enter a reference type.		E
AR_RAPI_ACTIVITY_X_INVALID	The specified combination of payment schedule identifier and receivables transaction identifier is invalid.	The activity type derived from the receivables_trx_id does not match with the activity type of the specified payment_schedule_id.	E
AR_RAPI_AMT_APP_FROM_INV_ALID	The allocated receipt amount and the applied amount should be same for the functional currency receipt.		E
AR_RAPI_APP_PS_ID_INVALID	Applied payment schedule identifier has an invalid value.		E
AR_RAPI_APP_PS_RA_ID_X_INV_ALID	Invalid receivable application identifier for the specified applied payment schedule identifier.		E
AR_RAPI_APPLIED_AMT_NULL	Applied amount could not be defaulted.	The p_applied_amount was not specified by the user and it could not be defaulted from the specified receipt or the specified transaction. For explanation on defaulting mechanism refer Defaulting, page 8-29	E
AR_RAPI_CASH_RCPT_ID_INVALID	Invalid cash receipt identifier.		E
AR_RAPI_CASH_RCPT_ID_NULL	Cash receipt identifier is null.		E
AR_RAPI_CC_RATE_AMTS_INV_ALID	The entered combination of the applied amount, allocated amount and the cross currency rate is invalid.	This error is raised if the following condition is violated in the cross currency applications: p_trans_to_receipt_rate* p_amount_applied = p_amount_applied_from.	E

MESSAGE_CODE	MESSAGE_TEXT	EXPLANATION	TYPE
AR_RAPI_CC_RATE_INVALID	Do not enter the cross currency rate if the receipt and the transaction are in same currency.	For the same currency receipt application, p_trans_to_receipt_rate should not be specified.	E
AR_RAPI_CC_RATE_NULL	Cross currency rate is null.	In case of a cross currency receipt application, the p_trans_to_receipt_rate could neither be defaulted nor derived.	E
AR_RAPI_CURR_CODE_INVALID	Currency code is invalid.	The specified currency code has an invalid value.	E
AR_RAPI_CUS_BK_AC_2_INVALID	Invalid combination of customer bank account name and number.	The specified combination of the p_customer_bank_account_number and p_customer_bank_account_name is invalid and cannot be used to derive the p_customer_bank_account_id.	E
AR_RAPI_CUS_BK_AC_ID_INVALID	Customer bank account identifier is invalid.	The specified value of p_customer_bank_account_id is invalid.	E
AR_RAPI_CUS_BK_AC_NAME_INVALID	Customer bank account name is invalid.	The specified value of p_customer_bank_account_name is invalid.	E
AR_RAPI_CUS_BK_AC_NUM_INVALID	Customer bank account number is invalid.	The specified value of p_customer_bank_account_number is invalid.	E
AR_RAPI_CUS_LOC_INVALID	Customer location is invalid for the specified customer.	The specified value of p_location has an invalid value.	E
AR_RAPI_CUS_NAME_INVALID	Invalid customer name.		E
AR_RAPI_CUS_NAME_NUM_INVALID	Invalid combination of customer name and number.		E
AR_RAPI_CUS_NUM_INVALID	Invalid customer number.		E

MESSAGE_CODE	MESSAGE_TEXT	EXPLANATION	TYPE
AR_RAPI_CUS_SITE_USE_ID_INV ALID	Customer site use identifier is invalid for the specified customer.	The specified value of p_customer_site_use_id is invalid for the given customer. It should be a valid BILL_TO site_use_id for the customer.	E
AR_RAPI_CUS_STE_USE_ID_NOT _DEF	Location could not be defaulted for the specified customer.	Neither the user had passed in any value for the p_location / p_customer_site_use_id, nor could it be defaulted to the primary Bill_To location for the given customer.	W
AR_RAPI_CUST_ID_INVALID	Customer identifier is invalid.		E
AR_RAPI_CUST_ID_NULL	Customer identifier is null.	The p_customer_id is null. For details, refer to API Usage, page 8-2.	E
AR_RAPI_CUS_NAME_NUM_IG N	Customer identifier has taken a precedence over name and number.	The specified values of p_customer_number and/or p_customer_name are ignored if the value for p_customer_id has been passed in.	W
AR_RAPI_CUST_TRX_ID_INVALID	Invalid customer transaction identifier.		E
AR_RAPI_CUST_TRX_ID_NULL	Customer transaction identifier is null.		E
AR_RAPI_DEF_TAX_FLAG_INVAL ID	Invalid deferred tax flag.	The valid values are 'Y'/'N'	E
AR_RAPI_DESC_FLEX_INVALID	The entered values for the descriptive flexfield &DFF_NAME is invalid.		E
AR_RAPI_DOC_SEQ_AUTOMATI C	You have passed in the document sequence value, even though the current document sequence is automatic.		E

MESSAGE_CODE	MESSAGE_TEXT	EXPLANATION	TYPE
AR_RAPI_DOC_SEQ_NOT_EXIST_A	Document sequence does not exist for the current document even though profile option Sequential Numbering is set to Always Used.		E
AR_RAPI_DOC_SEQ_NOT_EXIST_P	Document sequence does not exist for the current document even though profile option Sequential Numbering is set to Partially Used.		W
AR_RAPI_DOC_SEQ_VAL_INVALID	Document sequence value should not be entered if profile option Sequential Numbering is set to Not Used.		E
AR_RAPI_DOC_SEQ_VALUE_NULL_A	The profile option Sequential Numbering is set to Always Used and the document sequence is manual. The document sequence value is null.		E
AR_RAPI_DOC_SEQ_VALUE_NULL_P	The profile option Sequential Numbering is set to Partially Used and the document sequence is manual. The document sequence value is null.		W
AR_RAPI_FUNC_CURR_DEFAULTED	Functional currency defaulted as the receipt currency.		W
AR_RAPI_INS_PS_NOT_DEF_CUS	The customer could not be defaulted from the applied payment schedule identifier and the installment.	This error is raised if the customer_id cannot be derived from the p_applied_payment_schedule_id and the p_installment specified in the create_and_apply routine.	E
AR_RAPI_INSTALL_NULL	The installment number is null.		E
AR_RAPI_INVALID_APP_REF	Please supply a valid application reference type.		E

MESSAGE_CODE	MESSAGE_TEXT	EXPLANATION	TYPE
AR_RAPI_INVALID_CLAIM_ID	A valid claim ID &CLAIM_ID does not exist for the specified receipt and amount.		E
AR_RAPI_INVALID_CLAIM_NUM	The claim is invalid. Please enter a different claim number.		E
AR_RAPI_INVALID_REF_REASON	Please supply a valid reference reason.		E
AR_RAPI_MULTIPLE_ON_AC_APP	More than one On Account application exists for the current receipt. Please specify the receivable application identifier.	This error is raised in the unapply_on_account routine if for the specified cash receipt, more than one On Account application exists and the p_receivable_application_id is not specified.	E
AR_RAPI_NON_REVERSIBLE	Standard reversal not possible for this receipt.	Explanation: refer to Defaulting, page 8-68.	E
AR_RAPI_PSID_NOT_DEF_CUS	The customer could not be defaulted from the applied payment schedule identifier.	This error is raised in the create_and_apply routine if the customer is not entered and cannot be derived from the specified p_applied_payment_schedule_id.	E
AR_RAPI_RCPT_AMOUNT_NULL	Receipt amount is null.	This is a required field in the create_cash and the create_and_apply routines.	E
AR_RAPI_RCPT_MD_ID_NULL	Receipt method identifier is null.		E
AR_RAPI_RCPT_MD_NAME_IGN	Receipt method identifier has taken precedence over receipt method name.		W

MESSAGE_CODE	MESSAGE_TEXT	EXPLANATION	TYPE
AR_RAPI_RCPT_MD_NAME_INV ALID	Invalid receipt method name.	This error is raised if the p_receipt_method_id is not passed in and the specified p_receipt_method_name is invalid.	E
AR_RAPI_RCPT_NOT_APP_TO_I NV	There is no application of the entered receipt against the entered transaction.	This error is raised in the Unapply routine, if the specified receipt has no application against the specified transaction.	E
AR_RAPI_RCPT_NUM_IGN	Cash receipt identifier has taken a precedence over the receipt number.		W
AR_RAPI_RCPT_NUM_INVALID	Invalid receipt number.		E
AR_RAPI_RCPT_RA_ID_X_INVALID	Invalid combination of receivable application identifier and the cash receipt identifier.	The p_cash_receipt_id derived from the p_receivable_application_id specified by the user does not match with the p_cash_receipt_id which is either specified by the user or defaulted from the p_receipt_number.	E
AR_RAPI_RCT_MD_ID_INVALID	Invalid receipt method identifier.		E
AR_RAPI_RCPT_MD_NAME_INV ALID	Invalid receipt method name.		E
AR_RAPI_REC_APP_ID_INVALID	Invalid receivable application identifier.		E
AR_RAPI_REC_APP_ID_NULL	Receivable application identifier is null.	BR	E
AR_RAPI_REC_TRX_ID_INVALID	Invalid receivable transaction identifier.		E

MESSAGE_CODE	MESSAGE_TEXT	EXPLANATION	TYPE
AR_RAPI_Rem_BK_AC_2_Invalid_ID	Invalid combination of remittance bank account name and number.	The specified combination of the p_remittance_bank_account_number and p_remittance_bank_account_name is invalid, and cannot be used to derive the p_remittance_bank_account_id.	E
AR_RAPI_Rem_BK_AC_ID_Invalid_ID	Invalid remittance bank account identifier.	This error is raised if the specified p_remittance_bank_account_id is not associated with the specified p_receipt_method_id.	E
AR_RAPI_Rem_BK_AC_ID_Null	Remittance bank account identifier is null.		E
AR_RAPI_Rem_BK_AC_Name_Invalid	Invalid remittance bank account name.		E
AR_RAPI_Rem_BK_AC_Name_Invalid_IGN	Remittance bank account identifier has taken a precedence over the remittance bank account name and number.		W
AR_RAPI_Rem_BK_AC_Num_Invalid	Invalid remittance bank account number.		E
AR_RAPI_Rev_Cat_CD_Invalid	Invalid reversal category code.		E
AR_RAPI_Rev_Cat_CD_Null	Reversal category code is null.		E
AR_RAPI_Rev_Cat_Name_Invalid	Reversal category code has taken precedence over the reversal category name.		W
AR_RAPI_Rev_Cat_Name_Invalid_ID	Invalid reversal category name.		E
AR_RAPI_Rev_GL_Date_Null	Reversal GL date is null.		E

MESSAGE_CODE	MESSAGE_TEXT	EXPLANATION	TYPE
AR_RAPI_REV_REAS_CD_INVAL_ID	Invalid reversal reason code.		E
AR_RAPI_REV_REAS_CD_NULL	Reversal reason code is invalid.		E
AR_RAPI_REV_REAS_NAME_IN_N	Reversal reason code has taken a precedence over the reversal reason name.		W
AR_RAPI_REV_REAS_NAME_IN_VALID	Invalid reversal reason name.		E
AR_RAPI_TRX_ID_INST_INVALI_D	Invalid combination of the customer transaction identifier and installment.		E
AR_RAPI_TRX_INS_NOT_DEF_CUS	The customer could not be defaulted from the entered transaction and the installment.	This error is raised in the create_and_apply routine if the customer is not entered and cannot be derived from the specified transaction and installment.	E
AR_RAPI_TRX_INS_PS_NOT_DEF_CUS	The customer could not be defaulted from the entered transaction, installment and applied payment schedule identifier.	This error is raised in the create_and_apply routine if the customer is not entered and cannot be derived from the specified p_customer_trx_id/trx_number, p_installment and p_applied_payment_schedule_id.	E
AR_RAPI_TRX_LINE_AMT_DEFL_T	Amount applied has been defaulted to the line amount of the specified transaction line.		W
AR_RAPI_TRX_LINE_ID_INVALI_D	Invalid customer transaction line identifier.		E
AR_RAPI_TRX_LINE_NO_INVAL_ID	Invalid transaction line number.		E

MESSAGE_CODE	MESSAGE_TEXT	EXPLANATION	TYPE
AR_RAPI_TRX_NOT_DEF_CUST	The customer could not be defaulted from the entered transaction.	This error is raised in the create_and_apply routine if the customer is not entered and cannot be derived from the specified p_customer_trx_id/trx_number.	E
AR_RAPI_TRX_NUM_IGN	Customer transaction identifier has taken a precedence over the transaction number.		W
AR_RAPI_TRX_NUM_INST_INVALID	Invalid combination of transaction number and installment.		E
AR_RAPI_TRX_NUM_INVALID	Invalid transaction number.		E
AR_RAPI_TRX_PS_ID_X_INVALID	Invalid applied payment schedule identifier for the specified transaction.	The p_applied_payment_schedule_id specified by the user does not match with the payment_schedule_id derived from the p_customer_trx_id and the p_installment.	E
AR_RAPI_TRX_PS_NOT_DEF_CUSTOMER	The customer could not be defaulted from the entered transaction and the applied payment schedule identifier.	This error is raised in the create_and_apply routine if the customer is not entered and cannot be derived from the specified p_customer_trx_id/trx_number and the p_applied_payment_schedule_id.	E
AR_RAPI_TRX_RA_ID_X_INVALID	The activity type for the entered receivable transaction identifier does not match with the activity of the entered payment schedule identifier.	This message is to be used by the API, activity_application, added as part of the Bills Receivables changes.	E
AR_RAPI_USR_CURR_CODE_INVALID	Currency code took a precedence over the user currency code.		W

MESSAGE_CODE	MESSAGE_TEXT	EXPLANATION	TYPE
AR_RAPI_USR_CURR_CODE_INV ALID	User currency code is invalid.		E
AR_RAPI_USR_X_RATE_TYP_IN VALID	User exchange rate type is invalid.		E
AR_RAPI_USR_X_RATE_TYPE_IG N	Exchange rate type took a precedence over the User exchange rate type.		W
AR_RAPI_X_RATE_DATE_INVAL ID	Invalid exchange rate date.		E
AR_RAPI_X_RATE_INVALID	Exchange rate should not be entered.	This would be raised if the exchange rate type is not 'User' and the exchange rate has been specified.	E
AR_RAPI_X_RATE_NULL	Exchange rate is null.		E
AR_RAPI_X_RATE_TYPE_INVALI D	Invalid exchange rate type.		E
AR_RAPI_X_RATE_TYPE_NULL	Exchange rate type is null.		E
AR_RW_AMOUNT_LESS_THAN_ APP	The receipt amount cannot be less than the sum of the applied and on-account amounts.		E
AR_RW_APP_NEG_ON_ACCT	Amount applied cannot be negative for an On Account application.		E
AR_RW_APP_NEG_UNAPP	You may not apply more than the receipt amount.	This error is raised if you try to apply more than the unapplied amount on the receipt against a transaction.	E

MESSAGE_CODE	MESSAGE_TEXT	EXPLANATION	TYPE
AR_RW_APPLIED_GREATER_LINE	Amount applied cannot be greater than the original line amount of &AMOUNT.	This error is raised in the apply and create_and_apply routines if the line number of transaction has been specified and the amount applied is greater than the original line amount of the transaction line.	E
AR_RW_BEFORE_APP_GL_DATE	Reversal GL Date must be on or after original GL Date of &GL_DATE.		E
AR_RW_BEFORE_RECEIPT_GL_DATE	The Reversal GL Date cannot be before the Receipt GL Date.		E
AR_RW_CASH_DUPLICATE_RECEIPT	A cash receipt with this number, date, amount and customer already exists.		E
AR_RW_CC_RATE_POSITIVE	Cross currency rate must be greater than zero.	This error is raised in the apply and create_and_apply routines if the p_trans_to_receipt_rate has a negative value.	E
AR_RW_GL_DATE_BEFORE_RECEIPT_GL	The GL date cannot be before the receipt GL date.	This error is raised in the apply and the create_and_apply routines if the apply gl_date is before the receipt gl_date.	E
AR_RW_GL_DATE_BEFORE_OPEN_RECEIPT_GL	The application GL date must be later than the open receipt GL date for a receipt-to-receipt application.		E
AR_RW_MAT_BEFORE_RCT_DATE	The Maturity Date cannot be before the Receipt Date.		E
AR_RW_NET_DIFF_RCT_CURR	Both receipts in a receipt to receipt application must have the same currency.		E

MESSAGE_CODE	MESSAGE_TEXT	EXPLANATION	TYPE
AR_RW_NET_OPEN_AMT_INC	A receipt-to-receipt application must decrease the open receipt balance or bring the receipt balance closer to zero.		E
AR_RW_NET_OPEN_RCT_ONLY	Netting is allowed on open receipts only (unapplied cash, on-account cash and claim investigation applications).		E
AR_RW_NET_UNAPP_OVERAPP	Unapplying this payment netting application is not allowed because it would cause the applied receipt balance to become negative.		
AR_RW_NO_DISCNT	Discounts are not permitted for transactions with a negative original balance.		E
AR_RW_PAID_INVOICE_TWICE	You have paid the same invoice twice. Please correct.		E
AR_RW_RCT_AMOUNT_NEGATIVE	You cannot enter a negative receipt amount for cash receipts.		E
AR_RW_VAL_DISCOUNT	Discount taken is greater than the discount available (&DISC_AVAILABLE).		E
AR_RW_VAL_NEG_DISCNT	Discount cannot be negative.		E
AR_RW_VAL_ONACC_DISC	Discount not allowed for On Account application. Clear discount amount field or enter zero.		E
AR_RW_VAL_UNEARNED_DISCOUNT	Cannot take unearned discount because the Allow Unearned Discount system option is set to No.		E

MESSAGE_CODE	MESSAGE_TEXT	EXPLANATION	TYPE
AR_SYSTEM_WR_NO_LIMIT_SET	Please set the receipt write-off limits range system option.		E
AR_VAL_GL_INV_GL	The GL date should not be prior to the invoice's GL date.		E
AR_WR_NO_LIMIT	User Write-off limit does not exist.		E
AR_WR_TOTAL_EXCEED_MAX_AMOUNT	The total write-off amount must fall within the receipt write-off limits range system option.		E
AR_WR_USER_LIMIT	Total write-off amount must be in the range of &FROM_AMOUNT to &TO_AMOUNT.		E

Revenue Adjustment API User Notes

Overview

This document outlines the specifications and the methodology for using the various Revenue Adjustment APIs. These APIs provide an extension to existing functionality of adjusting revenue and sales credits through the standard AR Revenue Management form.

You can access these APIs:

- As standard PL/SQL servers-side routine calls
- Through Forms, utilizing the capability of Forms6 to have a procedure as its underlying base table

Basic Business Needs

The Revenue Adjustment API addresses the following basic functionality via different API calls:

- Unearning revenue
- Earning revenue
- Transferring sales credits between salespersons
- Adding new non-revenue sales credits

Presently, the main business need for the API is the requirement to have event-based revenue recognition. In Receivables, you can defer revenue recognition, and earn the revenue at a later date using the API. Throughout the process, the API uses AutoAccounting to determine the accounts to be debited/credited with each operation.

API Usage

To earn and unearn revenue, transfer sales credits, and add non-revenue sales credits at the transaction, item, category, or transaction line level, you can call the following four PL/SQL APIs:

- AR_RevenueAdjust_PUB.Unearn_Revenue, page 9-2: Transfers the specified amount of revenue from the revenue account to the unearned revenue account on the specified transaction lines.
- AR_RevenueAdjust_PUB.Earn_Revenue, page 9-16: Transfers the specified amount of revenue from the unearned revenue account to the revenue account on the specified transaction lines.
- AR_RevenueAdjust_PUB.Transfer_Sales_Credits, page 9-18: Transfers revenue and/or non-revenue sales credits between salespersons on the specified transaction lines. In the case of revenue sales credits, the associated revenue is also transferred between cost centers, assuming that AutoAccounting derives the cost center segment of the accounting flexfield from the salesperson.
- AR_RevenueAdjust_PUB.Add_Non_Revenue_Sales_Credits, page 9-23: Adds non-revenue sales credits for any salesperson to the specified transaction lines.
- AR_Revenueadjust_PUB.Record_Customer_Acceptance, page 9-27: Removes customer acceptance contingencies and enables revenue recognition.
- AR_Revenueadjust_PUB.Update_Contingency_Expirations, page 9-28: Updates the expiration date and period for specific time-based contingencies.

For all options, a specific amount or percentage of the total value can be specified. All available revenue can also be specified, except for Add_Non_Revenue_Sales_Credits, where this is not applicable.

Note: You cannot specify *both* revenue and nonrevenue sales credits when passing sales group information to the above APIs.

AR_RevenueAdjust_PUB.Unearn_Revenue

Call this routine to move revenue from the earned revenue account to the unearned revenue account using AutoAccounting. This API routine has 4 input and 5 output parameters in total. One of the input parameters is a record type that holds all the revenue adjustment information and has 120 elements. The output parameters include the revenue_adjustment_number and revenue_adjustment_id of the revenue adjustment.

The following is the breakdown of the parameters:

Input

Standard API parameters: 3

Revenue Adjustment parameters: 1 (revenue adjustment record type)

Output

Standard API parameters: 3

Revenue Adjustment parameters: 2

Parameter Descriptions

The input revenue adjustment parameter is a record of type
AR_Revenue_Adjustment_PVT.Rev_Adj_Rec_Type.

, FROM_ITEM_SEGMENT3	VARCHAR2 (40)
, FROM_ITEM_SEGMENT4	VARCHAR2 (40)
, FROM_ITEM_SEGMENT5	VARCHAR2 (40)
, FROM_ITEM_SEGMENT6	VARCHAR2 (40)
, FROM_ITEM_SEGMENT7	VARCHAR2 (40)
, FROM_ITEM_SEGMENT8	VARCHAR2 (40)
, FROM_ITEM_SEGMENT9	VARCHAR2 (40)
, FROM_ITEM_SEGMENT10	VARCHAR2 (40)
, FROM_ITEM_SEGMENT11	VARCHAR2 (40)
, FROM_ITEM_SEGMENT12	VARCHAR2 (40)
, FROM_ITEM_SEGMENT13	VARCHAR2 (40)
, FROM_ITEM_SEGMENT14	VARCHAR2 (40)
, FROM_ITEM_SEGMENT15	VARCHAR2 (40)
, FROM_ITEM_SEGMENT16	VARCHAR2 (40)
, FROM_ITEM_SEGMENT17	VARCHAR2 (40)
, FROM_ITEM_SEGMENT18	VARCHAR2 (40)
, FROM_ITEM_SEGMENT19	VARCHAR2 (40)
, FROM_ITEM_SEGMENT20	VARCHAR2 (40)
, TO_INVENTORY_ITEM_ID	NUMBER (15)
, TO_ITEM_SEGMENT1	VARCHAR2 (40)
, TO_ITEM_SEGMENT2	VARCHAR2 (40)
, TO_ITEM_SEGMENT3	VARCHAR2 (40)
, TO_ITEM_SEGMENT4	VARCHAR2 (40)
, TO_ITEM_SEGMENT5	VARCHAR2 (40)
, TO_ITEM_SEGMENT6	VARCHAR2 (40)
, TO_ITEM_SEGMENT7	VARCHAR2 (40)
, TO_ITEM_SEGMENT8	VARCHAR2 (40)
, TO_ITEM_SEGMENT9	VARCHAR2 (40)
, TO_ITEM_SEGMENT10	VARCHAR2 (40)
, TO_ITEM_SEGMENT11	VARCHAR2 (40)
, TO_ITEM_SEGMENT12	VARCHAR2 (40)
, TO_ITEM_SEGMENT13	VARCHAR2 (40)
, TO_ITEM_SEGMENT14	VARCHAR2 (40)
, TO_ITEM_SEGMENT15	VARCHAR2 (40)
, TO_ITEM_SEGMENT16	VARCHAR2 (40)
, TO_ITEM_SEGMENT17	VARCHAR2 (40)
, TO_ITEM_SEGMENT18	VARCHAR2 (40)
, TO_ITEM_SEGMENT19	VARCHAR2 (40)
, TO_ITEM_SEGMENT20	VARCHAR2 (40)
, FROM_CUST_TRX_LINE_ID	NUMBER (15)
, FROM_LINE_NUMBER	NUMBER (15)
, TO_CUST_TRX_LINE_ID	NUMBER (15)
, TO_LINE_NUMBER	NUMBER (15)
, GL_DATE	DATE
, REASON_CODE	VARCHAR2 (15)
, COMMENTS	VARCHAR2 (2000)
, ATTRIBUTE_CATEGORY	VARCHAR2 (30)
, 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)
, ATTRIBUTE11	VARCHAR2 (150)
, ATTRIBUTE12	VARCHAR2 (150)
, ATTRIBUTE13	VARCHAR2 (150)
, ATTRIBUTE14	VARCHAR2 (150)

```
,ATTRIBUTE15          VARCHAR2(150));
```

The following table lists standard API parameters that are common to all the routines in the Revenue Adjustment API.

Parameter	Type	Data-type	Required	Default Value	Description
p_api_version	IN	NUMBER	Yes		Used to compare version numbers of incoming calls to its current version number. Unexpected error is raised if version incompatibility exists. In the current version of the API, you should pass in a value of 1.0 for this parameter.
p_init_msg_list	IN	VARCHAR2		FND_API.G_FALSE	Allows API callers to request that the API does initialization of the message list on their behalf.
p_commit	IN	VARCHAR2		FND_API.G_FALSE	Used by API callers to ask the API to commit on their behalf.
p_rev_adj_rec	IN	AR_Revenue _Adjustment _PVT.Rev_A dj_Rec_Type	Yes	See break- down below for individual elements	Revenue Adjustment record type
x_return_status	OUT	VARCHAR2			Represents the API overall return status. Detailed in Return Status, page 1-4.
x_msg_count	OUT	NUMBER			Number of messages in the API message list.
x_msg_data	OUT	VARCHAR2			This is the message in encoded format if x_msg_count=1.
x_adjustment_id	OUT	NUMBER			The ID of the resulting revenue adjustment.
x_adjustment_n umber	OUT	VARCHAR2			The user visible number of the resulting revenue adjustment.

The following table lists Rev_Adj_Rec_Type elements that are relevant to Unearn_Revenue:

Note: If required parameters are not passed in a call to this API, then the call will fail. However, depending on the business scenario, you will have to pass in values for other parameters to successfully create the business object. Otherwise, error messages will be reported.

At least one of the numbered sets of parameters is required.

Parameter	Data-type	Required	Description
p_customer_trx_id	NUMBER(15)	1	<p>The ID of the transaction on which revenue is to be adjusted.</p> <p>Default: None</p> <p>Validation: Must exist if specified. Must not have a class of 'CB','DM','BR','DEP','GUAR' (i.e. chargeback, debit memo, bills receivable, deposit, guarantee). Must not have had credit memo(s) raised against the full transaction value. Warning if partial credit memo has been raised. Every line must have revenue sales credits adding to 100%.</p> <p>Errors: AR_TAPI_TRANS_NOT_EXIST AR_TW_INCORRECT_SALESREDIT AR_RA_CB_DISALLOWED AR_RA_DM_DISALLOWED AR_RA_BR_DISALLOWED AR_RA_DEP_DISALLOWED AR_RA_GUAR_DISALLOWED AR_TW_INCORRECT_SALESREDIT AR_RA_FULL_CREDIT</p> <p>Warnings: AR_RA_PARTIAL_CREDIT</p>
trx_number	ra_customer_trx trx_ 1 number%TYPE	1	<p>The user visible transaction number</p> <p>Default: None</p> <p>Validation: Ignored if customer_trx_id has a value. Must be unique. Batch source can be optionally passed as extra assurance of uniqueness - then must be unique for that batch source. Otherwise, validation is the same as for customer_trx_id.</p> <p>Errors: AR_RA_TRX_NOTFOUND AR_RA_TRX_TOO_MANY_ROWS</p>

Parameter	Data-type	Required	Description
batch_source_name	ra_batch_sources.name%TYPE		<p>Name of the batch source associated with the trx_number, if specified. Only used in association with trx_number to help ensure uniqueness.</p> <p>Default: None</p> <p>Validation: Ignored if trx_number is not passed. If an invalid string is passed, the trx not found message will result.</p>
adjustment_type	VARCHAR2(15)		<p>Type of revenue adjustment. This element should be left null.</p> <p>Default: 'UN'</p>
from_salesrep_id	NUMBER(15)		<p>The ID of the salesperson whose revenue is being adjusted.</p> <p>Validation: If specified, must exist, must be currently active, and must have been active on transaction date. Must have revenue sales credits on at least one line on the transaction.</p> <p>Error: AR_TAPI_INVALID_SALESREP_ID AR_RA_SALESREP_NOT_ON_TRX</p>
from_salesrep_number	ra_salesreps.salesrep_number%TYPE		<p>The user visible number of the salesperson whose revenue is being adjusted.</p> <p>Validation: Ignored if from_salesrep_id is specified. Otherwise, validation is as for from_salesrep_id.</p> <p>Error: AR_RA_INAVLID_SALESREP_NUMBER</p>
to_salesrep_id	NUMBER		<p>Not used for unearning revenue and should be left null.</p>
to_salesrep_number	VARCHAR2(30)		<p>Not used for unearning revenue and should be left null.</p>

Parameter	Data-type	Required	Description
from_salesgroup_id	jtf_rs_groups_b. group_id%TYPE		<p>The ID of the sales group of the salesperson whose revenue is being adjusted.</p> <p>Validation: Must have revenue sales credits on at least one line on the transaction.</p> <p>Error: AR_RA_SALESREP_NOT_ON_TRX</p>
to_salesgroup_id	jtf_rs_groups_b. group_id%TYPE		<p>Not used for unearning revenue and should be left null.</p>
sales_credit_type	VARCHAR2(15)		<p>Not used for unearning revenue and should be left null.</p>
amount_mode	VARCHAR2(15)		<p>The amount mode specifies whether an amount, a percentage (of total value of selected lines), or all adjustable revenue is to be adjusted. Possible values are:</p> <ul style="list-style-type: none"> • T - total adjustable revenue • A - amount • P - percent <p>Default: 'T'</p> <p>Validation: Must be one of the above values</p> <p>Error: AR_RA_INVALID_AMOUNT_MODE</p>
amount	NUMBER		<p>The amount of revenue to be adjusted</p> <p>Default: None</p> <p>Validation: Ignored unless amount_mode = 'A', in which case it must have a value. Must be <= total recognized revenue for selected lines, and salesperson (if specified).</p> <p>Errors:</p> <p>AR_RA_AMT_EXCEEDS_AVAIL_REV AR_RA_ZERO_AMOUNT</p>

Parameter	Data-type	Required	Description
percent	NUMBER		<p>The percentage of total selected transaction line value to be adjusted.</p> <p>Default: None</p> <p>Validation: Ignored unless amount_mode = 'P' in which case it must have a value. Must be <= percentage of total value of selected lines represented by recognized revenue for selected lines, and salesperson (if specified).</p> <p>Errors: AR_RA_PCT_EXCEEDS_AVAIL_PCT AR_RA_ZERO_AMOUNT</p>
line_selection_mode	VARCHAR2(15)		<p>The line selection mode determines how lines were selected for adjustment.</p> <p>Possible values are:</p> <ul style="list-style-type: none"> • A - All transaction lines • C - Specific category • I - Specific item • S - Specific line. <p>Default: 'A'</p> <p>Validation: Must be one of the above values</p> <p>Error: AR_RA_INVALID_LINE_MODE</p>
from_category_id	NUMBER(15)		<p>The ID of the item category used to identify the lines to be adjusted.</p> <p>Default: None</p> <p>Validation: Must be a valid category ID, and there must be lines on the transaction that have items belonging to this category. Must be specified if line selection mode = 'C'.</p> <p>Errors: AR_RA_NO_FROM_CATEGORY AR_RA_INVALID_CATEGORY_ID AR_RA_CATEGORY_NOT_ON_TRX</p>

Parameter	Data-type	Required	Description
from_category_segment1 -from_category_segment2 0	VARCHAR2(40)		Segments 1 to 20 of the category flexfield Default: None
			Validation: Ignored if from_category_id has a value. Enough segment values to uniquely identify a category must be passed - ideally all defined segments. Otherwise, validation is the same as for from_category_id. Error: AR_RA_INVALID_CAT_SEGMENTS
to_category_id	NUMBER(15)		Not currently used and should be left null.
to_category_segment1 -to_category_segment20	VARCHAR2(40)		Not currently used and should be left null.
from_inventory_item_id	NUMBER(15)		The ID of the inventory item used to identify the lines to be adjusted. Default: None
			Validation: Must be a valid inventory item ID and there must be lines on the transaction that have items with this ID. Must be specified if line selection mode = 'T'. Errors: AR_RA_NO_FROM_ITEM AR_RA_INVALID_ITEM_ID AR_RA_ITEM_NOT_ON_TRX
from_item_segment1 -from_item_segment20	VARCHAR2(40)		Segments 1 to 20 of the item flexfield Default: None
			Validation: Ignored if from_inventory_item_id has a value. Enough segment values to uniquely identify an item must be passed - ideally all defined segments. Otherwise, validation is the same as for from_inventory_item_id. Error: AR_RA_INVALID_ITEM_SEGMENTS
to_inventory_item_id	NUMBER(15)		Not currently used and should be left null.

Parameter	Data-type	Required	Description
to_item_segment1 -to_item_segment20	VARCHAR2(40)		Not currently used and should be left null.
from_cust_trx_line_id	NUMBER(15)		<p>The ID of the transaction line to be adjusted.</p> <p>Default: None</p> <p>Validation: Must be a valid line ID on the transaction. Must be specified if line selection mode = 'S' and from_line_number is null.</p> <p>Errors: AR_RA_NO_FROM_LINE AR_RA_INVALID_LINE_ID</p>
from_line_number	NUMBER(15)		<p>The user visible transaction line number.</p> <p>Default: None</p> <p>Validation: Ignored if from_cust_trx_line_id has a value. Must be a valid line number on the transaction.</p> <p>Errors: AR_RA_NO_FROM_LINE AR_RA_LINE_NOT_ON_TRX</p>

Parameter	Data-type	Required	Description
gl_date	DATE		<p>Date that adjusted revenue will be posted to the general ledger if revenue is recognized immediately. Start date of revenue recognition if revenue is deferred.</p> <p>Default: Gets defaulted to the current date if it is a valid gl_date.</p> <p>Validation: Ignored for lines that have non-deferred accounting rules AND a duration > 1. It is valid if the following conditions are true:</p> <ul style="list-style-type: none"> • The date is in an Open or Future period, or it is in a Never Opened period and the Allow Not Open Flag is set to Yes. • The date is greater than or equal to the trx_date • The period cannot be an Adjustment period. <p>If the date passed is not valid, then a warning message is written to the stack and the date is automatically overridden with a valid date using the default:</p> <ul style="list-style-type: none"> • If the most recent open period is prior to the current date: last date of that period • If there is a period open after the current date: first date of the last open period <p>Warning: AR_RA_GL_DATE_CHANGED</p>
reason_code	VARCHAR2(15)	Yes	<p>Lookup code for revenue adjustment reason</p> <p>Default: None</p> <p>Validation: Must be defined under AR lookup type 'REV_ADJ_REASON'</p> <p>Error: AR_RA_INVALID_REASON_CODE</p>

Parameter	Data-type	Required	Description
comments	VARCHAR2 (2000)		Free text Default: None Validation: None
attribute_category	VARCHAR2(30)		Context of the revenue adjustment descriptive flexfield. Default: None Validation: None
attribute1 - attribute15	VARCHAR2(150)		Attributes of the revenue adjustment descriptive flexfield Default: None Validation: Standard descriptive flexfield validation

Example

Objective:

To unearn all revenue on a transaction using a call to *AR_RevenueAdjust_PUB.Unearn_Revenue* and passing a minimum number of input parameters.

This table lists the entered parameters:

Parameter	Entered Value	Default Value
p_api_version	2.0	
p_init_msg_list	FND_API.G_TRUE	
p_rev_adj_rec.trx_number	'test_api_1'	
p_rev_adj_rec.reason_code	'RA'	

This table lists the defaulted input parameters, which were not entered:

Parameter	Entered Value	Default Value
p_rev_adj_rec.amount_mode	'T'	
p_rev_adj_rec.line_selection_mode	'A'	
p_rev_adj_rec.gl_date	SYSDATE	

The API call in this case would be:

```
AR_RevenueAdjust_PUB.Unearn_Revenue(
    p_api_version      => 2.0,
    p_init_msg_list    => FND_API.G_TRUE,
    p_rev_adj_rec.trx_number  => 'test_api_1',
    p_rev_adj_rec.reason_code => 'RA',
    x_return_status     => l_return_status,
    x_msg_count         => l_msg_count,
    x_msg_data          => l_msg_data,
    x_adjustment_id    => l_adjustment_id,
    x_adjustment_number => l_adjustment_number);
```

After execution of this API, the calling program retrieves the warnings and the error messages, put on the message stack by the API, in the following manner:

The warnings and the error messages put on the message stack by the API are retrieved after the execution of this API by the calling program, in the following manner:

```
IF l_msg_count = 1 Then
    --there is one message raised by the API, so it has been sent out
    --in the parameter x_msg_data, get it.
    l_msg_data_out := l_msg_data;
ELSIF l_msg_count > 1 Then
    --the messages on the stack are more than one so call them in a loop
    -- and put the messages in a PL/SQL table.
    loop
        count := count +1 ;
        l_mesg := FND_MSG_PUB.Get;
        If l_mesg IS NULL Then
            EXIT;
        else
            Mesg_tbl(count).message := l_mesg;
        End if;
    end loop;
END IF;
```

Depending on the message level threshold set by the profile option FND_API_MSG_LEVEL_THRESHOLD, the messages put on the message stack may contain both the error messages and the warnings.

Result:

All revenue on this transaction was unearned by specifying only four input parameters in the call to this API.

AR_RevenueAdjust_PUB.Earn_Revenue

Call this routine to move revenue from the unearned revenue account to the earned revenue account using AutoAccounting. This API routine has 4 input and 5 output parameters in total and is almost exactly the same as the `Unearn_Revenue` routine described above in `AR_RevenueAdjust_PUB.Unearn_Revenue`, page 9-2.

The following is the breakdown of the parameters:

Input

Standard API parameters: 3

Revenue Adjustment parameters: 1 (revenue adjustment record type)

Output

Standard API parameters: 3

Revenue Adjustment parameters: 2

Parameter Descriptions

For a description of this routine's standard parameters, see `AR_RevenueAdjust_PUB.Unearn_Revenue`, page 9-2.

The `Rev_Adj_Rec_Type` elements that are relevant to `Earn_Revenue` are exactly the same as already listed in `AR_RevenueAdjust_PUB.Unearn_Revenue`, with the following exceptions listed in this table:

Parameter	Data-type	Required	Description
<code>to_salesrep_id</code>	NUMBER		Not used for earning revenue and should be left null.
<code>to_salesrep_number</code>	VARCHAR2		Not used for earning revenue and should be left null.
<code>to_salesgroup_id</code>	<code>jtf_rs_groups_b.</code> <code>group_id%TYPE</code>		Not used for earning revenue and should be left null.
<code>sales_credit_type</code>	VARCHAR2(15)		Not used for earning revenue and should be left null.

Example

Objective:

To earn all revenue on a transaction using a call to *AR_RevenueAdjust_PUB.Earn_Revenue* and passing a minimum number of input parameters.

This table lists the entered parameters:

Parameter	Entered Value	Default Value
p_api_version	2.0	
p_init_msg_list	FND_API.G_TRUE	
p_rev_adj_rec trx_number	'test_api_1'	
p_rev_adj_rec.reason_code	'RA'	

This table lists the defaulted input parameters, which were not entered:

Parameter	Entered Value	Default Value
p_rev_adj_rec.amount_mode		'T'
p_rev_adj_rec.line_selection_mode		'A'
p_rev_adj_rec.gl_date		SYSDATE

The API call in this case would be:

```
AR_RevenueAdjust_PUB.Earn_Revenue(
    p_api_version          => 2.0,
    p_init_msg_list         => FND_API.G_TRUE,
    p_rev_adj_rec trx_number => 'test_api_1',
    p_rev_adj_rec.reason_code => 'RA',
    x_return_status         => l_return_status,
    x_msg_count              => l_msg_count,
    x_msg_data                => l_msg_data,
    x_adjustment_id          => l_adjustment_id,
    x_adjustment_number       => l_adjustment_number);
```

The warnings and the error messages put on the message stack by the API are retrieved after the execution of this API by the calling program, as described in Example, page 9-14.

Result:

All revenue on this transaction was earned by specifying only four input parameters in the call to this API.

AR_RevenueAdjust_PUB.Transfer_Sales_Credits

Call this routine to transfer sales credits from any salesperson with sales credits on the transaction to any other salesperson. In addition, if revenue sales credits are transferred, then the associated revenue is transferred between cost centers if the AutoAccounting rules call the salesperson table and the cost center segment is derived from the salesperson.

This API routine has 4 input and 5 output parameters in total and is similar to the `Unearn_Revenue` routine described above in

`AR_RevenueAdjust_PUB.Unearn_Revenue`, page 9-2. The following is the breakdown of the parameters:

Input

Standard API parameters: 3

Revenue Adjustment parameters: 1 (revenue adjustment record type)

Output

Standard API parameters: 3

Revenue Adjustment parameters: 2

Parameter Descriptions

For a description of this routine's standard parameters, see `AR_RevenueAdjust_PUB.Unearn_Revenue`, page 9-2.

The `Rev_Adj_Rec_Type` elements that are relevant to `Transfer_Sales_Credits` are the same as already listed in `AR_RevenueAdjust_PUB.Unearn_Revenue`, with the following exceptions/additions listed in this table.

Note: If required parameters are not passed in a call to this API, then the call will fail. However, depending on the business scenario, you will have to pass in values for other parameters to successfully create the business object. Otherwise, error messages will be reported.

Parameter	Data-type	Required	Description
p_customer_trx_id	NUMBER(15)	1	<p>The ID of the transaction on which sales credits are to be adjusted.</p> <p>Default: None</p> <p>Validation: Must exist if specified. Must not have a class of 'CB','BR','GUAR' (i.e. chargeback, bills receivable, guarantee). Must not have had credit memo(s) raised against the full transaction value. Warning if partial credit memo has been raised. Every line must have revenue sales credits adding to 100%.</p> <p>Errors: AR_TAPI_TRANS_NOT_EXIST AR_TW_INCORRECT_SALESCREDIT AR_RA_CB_DISALLOWED AR_RA_BR_DISALLOWED AR_RA_GUAR_DISALLOWED AR_TW_INCORRECT_SALESCREDIT AR_RA_FULL_CREDIT</p> <p>Warnings: AR_RA_PARTIAL_CREDIT</p>
from_salesrep_id	NUMBER(15)		<p>The ID of the salesperson from whom sales credits are being transferred.</p> <p>Default: Null</p> <p>Validation: If specified, must exist, must be currently active, and must have been active on transaction date. Must have revenue sales credits on at least one line on the transaction. If neither from_salesrep_id nor from_salesrep_number are specified, sales credits of the specified type are transferred belonging to all salesreps on the transaction (i.e. null = all).</p> <p>Error: AR_TAPI_INVALID_SALESPREP_ID AR_RA_SALESPREP_NOT_ON_TRX</p>

Parameter	Data-type	Required	Description
from_salesrep_number	ra_salesreps.sal esrep_number% TYPE		<p>The user visible number of the salesperson from whom sales credits are being transferred.</p> <p>Validation: Ignored if from_salesrep_id is specified. Otherwise, validation is as for from_salesrep_id.</p> <p>Error: AR_RA_INVALID_SALESREP_NUMBER</p>
to_salesrep_id	NUMBER(15) 2		<p>The ID of the salesperson to whom sales credits are being transferred.</p> <p>Validation: If specified, must exist, and must be currently active and must have been active on transaction date.</p> <p>Errors: AR_TAPI_INVALID_SALESREP_ID AR_RA_NO_TO_SALESREP</p>
to_salesrep_number	ra_salesreps.sal esrep_number% TYPE	2	<p>The user visible number of the salesperson to whom sales credits are being transferred.</p> <p>Validation: Ignored if to_salesrep_id is specified. Otherwise, validation is as for to_salesrep_id.</p> <p>Error: AR_RA_INVALID_SALESREP_NUMBER</p>
from_salesgroup_id	jtf_rs_groups_b. group_id%TYP E		<p>The ID of the sales group of the salesperson from whom sales credits are being transferred.</p> <p>Default: Null</p> <p>Validation: Must have sales credits (of the type being transferred) on at least one line on the transaction. If FROM_SALESGROUP_ID is not specified, then all sales credits of the specified type for the chosen salesperson are transferred (ie. null = all).</p> <p>Error: AR_RA_SALESREP_NOT_ON_TRX</p>

Parameter	Data-type	Required	Description
to_salesgroup_id	jtf_rs_groups_b. group_id%TYP E		<p>The ID of the sales group of the salesperson to whom sales credits are being transferred.</p> <p>Validation: If specified, then must exist and must be currently active. Salesperson must have been an active member of this group at some time between:</p> <ul style="list-style-type: none"> • the earliest of the transaction date and any parent commitment/invoice dates, and • the latest of the current date, transaction date, and any parent commitment/invoice dates. <p>Error: AR_INVALID_SALESGROUP_ID</p>
sales_credit_type	VARCHAR2(15)	Yes	<p>The type of sales credit being transferred. Possible values:</p> <ul style="list-style-type: none"> • R = revenue sales credits • N = non-revenue sales credits • B = both <p>Default: 'R'</p> <p>Validation: Must be one of the above values.</p> <p>Note: The value B cannot be used if either FROM_SALESGROUP_ID or TO_SALESGROUP_ID is specified.</p> <p>Error: AR_INCOMPATIBLE_CREDIT_TYPE AR_RA_INVALID_SALESCRED_TYPE</p>

Example

Objective:

To transfer all revenue sales credits on a transaction from a salesperson to a new salesperson using a call to *AR_RevenueAdjust_PUB.Transfer_Sales_Credits* and passing a minimum number of input parameters.

This table lists the entered parameters:

Parameter	Entered Value	Default Value
p_api_version	2.0	
p_init_msg_list	FND_API.G_TRUE	
p_rev_adj_rec trx_number	'test_api_1'	
p_rev_adj_rec from_salesrep_number	'101'	
p_rev_adj_rec to_salesrep_number	'299'	
p_rev_adj_rec.reason_code	'RA'	

This table lists the defaulted input parameters, which were not entered:

Parameter	Entered Value	Default Value
p_rev_adj_rec.amount_mode		'T'
p_rev_adj_rec.sales_credit_type		'R'
p_rev_adj_rec.line_selection_mode		'A'
p_rev_adj_rec.gl_date		SYSDATE

The API call in this case would be:

```
AR_RevenueAdjust_PUB.Transfer_Sales_Credits(
    p_api_version          => 2.0,
    p_init_msg_list        => FND_API.G_TRUE,
    p_rev_adj_rec trx_number => 'test_api_1',
    p_rev_adj_rec from_salesrep_number => '101',
    p_rev_adj_rec to_salesrep_number => '299',
    p_rev_adj_rec.reason_code => 'RA',
    x_return_status        => l_return_status,
    x_msg_count            => l_msg_count,
    x_msg_data             => l_msg_data,
    x_adjustment_id        => l_adjustment_id,
    x_adjustment_number    => l_adjustment_number);
```

The warnings and the error messages put on the message stack by the API are retrieved after execution of this API by the calling program, as described in Example, page 9-14.

Result:

All revenue sales credits on this transaction belonging to salesperson 101 were transferred to salesperson 299 by specifying only six input parameters in the call to this API. Additionally, all associated revenue was transferred between corresponding cost centers. Note that if salesrep number 101 was the only salesperson with revenue sales credits on this transaction, then `from_salesrep_number` could have been omitted. This is because no specified salesperson means *all* salespersons, thereby cutting the required number of parameters to five.

AR_RevenueAdjust_PUB.Add_Non_Revenue_Sales_Credits

Call this routine to add non-revenue sales credits to any existing or new salesperson on a transaction. This does not involve a transfer of revenue. This API routine has 4 input and 5 output parameters in total and is similar to the `Unearn_Revenue` routine described in `AR_RevenueAdjust_PUB.Unearn_Revenue`, page 9-2.

The following is the breakdown of the parameters:

Input

Standard API parameters: 3

Revenue Adjustment parameters: 1 (revenue adjustment record type)

Output

Standard API parameters: 3

Revenue Adjustment parameters: 2

Parameter Descriptions

For a description of this routine's standard parameters, see `AR_RevenueAdjust_PUB.Unearn_Revenue`, page 9-2.

The `Rev_Adj_Rec_Type` elements that are relevant to `Add_Non_Revenue_Sales_Credits` are the same as already listed in `AR_RevenueAdjust_PUB.Unearn_Revenue`, with the following exceptions/additions listed in this table:

Note: If required parameters are not passed in a call to this API, then the call will fail. However, depending on the business scenario, you will have to pass in values for other parameters to successfully create the business object. Otherwise, error messages will be reported.

At least one of the numbered sets of parameters is required.

Parameter	Data-type	Required	Description
p_customer_trx_id	NUMBER(15)	1	<p>The ID of the transaction on which sales credits are to be adjusted.</p> <p>Default: None</p> <p>Validation: Must exist if specified. Must not have a class of 'CB','BR','GUAR' (i.e. chargeback, bills receivable, guarantee). Must not have had credit memo(s) raised against the full transaction value. Warning if partial credit memo has been raised. Every line must have revenue sales credits adding to 100%.</p> <p>Errors:</p> <ul style="list-style-type: none"> AR_TAPI_TRANS_NOT_EXIST AR_TW_INCORRECT_SALESCREDIT_AR_RA_CB_DISALLOWED AR_RA_BR_DISALLOWED AR_RA_GUAR_DISALLOWED AR_TW_INCORRECT_SALESCREDIT_AR_RA_FULL_CREDIT <p>Warnings:</p> <ul style="list-style-type: none"> AR_RA_PARTIAL_CREDIT
from_salesrep_id	NUMBER(15)		Not applicable in this context and should be left null.
from_salesrep_number	ra_salesreps.salesrep_number%TYPE		Not applicable in this context and should be left null.
to_salesrep_id	NUMBER(15)	2	<p>The ID of the salesperson to whom non-revenue sales credits are being added.</p> <p>Validation: If specified, must exist, and must be currently active and must have been active on transaction date.</p> <p>Errors:</p> <ul style="list-style-type: none"> AR_TAPI_INVALID_SALESPREP_ID AR_RA_NO_TO_SALESPREP

Parameter	Data-type	Required	Description
to_salesrep_number	ra_salesreps.salesrep_number%TYPE	2	<p>The user visible number of the salesperson to whom sales credits are being transferred.</p> <p>Validation: Ignored if to_salesrep_id is specified. Otherwise, validation is as for to_salesrep_id.</p> <p>Error: AR_RA_INVALID_SALESREP_NUMBER</p>
from_salesgroup_id	jtf_rs_groups_b.group_id%TYPE		<p>Not applicable in this context and should be left null.</p>
to_salesgroup_id	jtf_rs_groups_b.group_id%TYPE		<p>The ID of the sales group of the salesperson to whom nonrevenue sales credits are being added.</p> <p>Validation: If specified, then must exist and must be currently active. Salesperson must have been an active member of this group at some time between:</p> <ul style="list-style-type: none"> the earliest of the transaction date and any parent commitment/invoice dates, and the latest of the current date, transaction date, and any parent commitment/invoice dates. <p>Error: AR_INVALID_SALESGROUP_ID</p>
sales_credit_type	VARCHAR2(15)		<p>Not applicable in this context and should be left null.</p>

Parameter	Data-type	Required	Description
amount_mode	VARCHAR2(15)		<p>The amount mode specifies whether an amount, a percentage (of total value of selected lines) is to be adjusted. Possible values are:</p> <ul style="list-style-type: none"> • A - amount • P - percent <p>Default: 'T', or all adjustable revenue is not applicable in this context.</p> <p>Validation: Must be one of the above values (A or P).</p> <p>Error: AR_RA_INVALID_AMOUNT_MODE</p>

Example

Objective:

To add 50% of the total transaction value in non-revenue sales credits to a new salesperson on a transaction, using a call to AR_RevenueAdjust_PUB.Add_Non_Revenue_Sales_Credits and passing a minimum number of input parameters.

This table lists the entered parameters:

Parameter	Entered Value	Default Value
p_api_version	2.0	
p_init_msg_list	FND_API.G_TRUE	
p_rev_adj_rec.trx_number	'test_api_1'	
p_rev_adj_rec.to_salesrep_number	'299'	
p_rev_adj_rec.amount_mode	'P'	

Parameter	Entered Value	Default Value
p_rev_adj_rec.percent	50	
p_rev_adj_rec.reason_code	'RA'	

This table lists the defaulted input parameters, which were not entered:

Parameter	Entered Value	Default Value
p_rev_adj_rec.line_selection_mode		'A'
p_rev_adj_rec.gl_date		SYSDATE

The API call in this case would be:

```
AR_RevenueAdjust_PUB.Add_Non_Revenue_Sales_Credits(
    p_api_version          => 2.0,
    p_init_msg_list         => FND_API.G_TRUE,
    p_rev_adj_rec.trx_number => 'test_api_1',
    p_rev_adj_rec.to_salesrep_number => '299',
    p_rev_adj_rec.amount_mode => 'P',
    p_rev_adj_rec.percent    => 50,
    p_rev_adj_rec.reason_code => 'RA',
    x_return_status          => l_return_status,
    x_msg_count              => l_msg_count,
    x_msg_data               => l_msg_data,
    x_adjustment_id          => l_adjustment_id,
    x_adjustment_number      => l_adjustment_number);
```

The warnings and the error messages put on the message stack by the API are retrieved after execution of this API by the calling program, as described in Example, page 9-14.

Result:

Non-revenue sales credits were added to salesperson 299 on this transaction by specifying only seven input parameters in the call to this API.

AR_Revenueadjust_PUB.Record_Customer_Acceptance

Call this routine to accept post-billing customer acceptance clauses in Receivables, and, if required, in Order Management, to remove these type of contingencies and to recognize revenue. This API routine has 5 input and 5 output parameters in total and is similar to the `Unearn_Revenue` routine described in `AR_RevenueAdjust_PUB.Unearn_Revenue`, page 9-2.

The following is the breakdown of the parameters:

Input

Standard API parameters: 3

Revenue Adjustment parameters: 1 (revenue adjustment record type)

Organization Identifier parameter: 1

Output

Standard API parameters: 3

Revenue Adjustment parameters: 2

Parameter Descriptions

For a description of this routine's standard parameters, see AR_RevenueAdjust_PUB.Unearn_Revenue, page 9-2.

The Rev_Adj_Rec_Type elements that are relevant to Record_Customer_Acceptance are exactly the same as already listed in AR_RevenueAdjust_PUB.Unearn_Revenue, with the following exceptions listed in this table:

Parameter	Data-type	Required	Description
p_org_id	jtf_rs_salesreps.or g_i d%type	Yes	Organization Identifier

AR_Revenueadjust_PUB.Update_Contingency_Expirations

Call this routine to update the expiration date and period for specific time-based contingencies. This API routine has 5 input and 5 output parameters in total and is almost exactly the same as the Unearn_Revenue routine described in AR_RevenueAdjust_PUB.Unearn_Revenue, page 9-2.

The following is the breakdown of the parameters:

Input

Standard API parameters: 3

Revenue Adjustment parameters: 1 (revenue adjustment record type)

Organization Identifier parameter: 1

Output

Standard API parameters: 3

Revenue Adjustment parameters: 2

Parameter Descriptions

For a description of this routine's standard parameters, see AR_RevenueAdjust_PUB.Unearn_Revenue, page 9-2.

The Rev_Adj_Rec_Type elements that are relevant to Update_Contingency_Expirations are exactly the same as already listed in AR_RevenueAdjust_PUB.Unearn_Revenue, with the following exceptions listed in this table:

Parameter	Data-type	Required	Description
p_org_id	jtf_rs_salesreps.org _i d%type	Yes	Organization Identifier.
p_customer_trx_id	ra_customer_trx.cu stomer_trx_id	Yes	Identifies the affected invoices; affects all contingencies for the transaction.
p_customer_trx_line_id	ra_customer_trx_li nes.customer_trx_l ine_id		Identifies the specific affected line and contingencies.
p_contingency_id	ar_line_consts.contn gency_id		Restricts updates to a specific type of contingency for the previously identified transaction or line.
p_expiration_date	ar_line_consts.expir ation_date		The new or resulting expiration date for the contingencies.
p_expiration_days	ar_line_consts.expir ation_days		The new offset for calculating the expiration date on the contingencies. The new expiration date is calculated as follows: Expiration Date = Expiration Event Date + Expiration Days

Example

Objective:

To update the expiration dates for all contingencies associated with a transaction to 31-DEC-2008, using a call to AR_Revenueadjust_PUB.Update_Contingency_Expirations and passing a minimum number of input parameters.

This table lists the input parameters:

Parameter	Entered Value	Default Value
p_api_version	2.0	
p_init_msg_list	FND_API.G_TRUE	
p_commit	FND_API.G_TRUE	
p_org_id	204	
p_customer_trx_id	1112234	
p_customer_trx_line_id	NULL	
p_contingency_id	NULL	
p_expiration_date	31-DEC-2008	
p_expiration_days	NULL	

Important: For updating contingency expirations, you must enter a value for either the p_expiration_date or the p_expiration_days parameter. Not providing a value for either of the parameters raises a validation error. If you provide both the values, the API honors the p_expiration_date first.

The API call in this case would be:

```
AR_RevenueAdjust_PUB.update_contingency_expirations(
    p_api_version          => 2.0,
    p_init_msg_list         => FND_API.G_TRUE,
    p_commit                => FND_API.G_TRUE,
    p_org_id                => 204,
    p_customer_trx_id       => 1112234
    p_expiration_date       => '31-DEC-2008'
    x_return_status          => l_return_status,
    x_msg_count              => l_msg_count,
    x_msg_data               => l_msg_data);
```

The warnings and the error messages put on the message stack by the API are retrieved after execution of this API by the calling program, as described in Example, page 9-14.

Result:

All existing contingencies associated with this transaction now have their expiration dates set to 31-DEC-2008. On the expiration date, provided no other activity has already

released the revenue, these contingencies will expire and revenue will be earned.

Messages

Messages play an important role in the effectiveness of API calls. The right message is raised at the right point to convey the exact error that has occurred or any warnings that have been raised. In the Revenue Adjustment API, all error messages and warnings raised during execution are put on the message stack and can be retrieved by the user as described in Exception Handling and Result Messages, page 1-3.

WARNINGS AND ERRORS

The following table lists all the error messages raised by the Revenue Adjustment API:

TYPE

E: Error message

W: Warning message

MESSAGE CODE	MESSAGE TEXT	DESCRIPTION	TYPE
AR_INCOMPATIBLE_CREDIT_TYPE	The option of transferring "both" sales credit types is not available in conjunction with sales group transfers.		E
AR_INVALID_SALESGROUP_ID	Please provide a valid sales group ID for sales credit transfers or additions.		E
AR_RA_AMT_EXCEEDS_AVAIL_REV	The amount entered is greater than &TOT_AVAIL_REV, the total available revenue on the lines selected	This message is generated by the revenue adjustment API when there is insufficient adjustable revenue on the selected transaction lines to meet the specified amount.	E
AR_RA_BR_DISALLOWED	Revenue cannot be adjusted on bills receivable		E
AR_RA_CATEGORY_NOT_ON_TRANSACTION	There are no lines with items for category ID &CATEGORY_ID on this transaction.		E

MESSAGE CODE	MESSAGE TEXT	DESCRIPTION	TYPE
AR_RA_CB_DISALLOWED	Revenue cannot be adjusted on chargebacks		E
AR_RA_DEP_DISALLOWED	Revenue cannot be adjusted on deposits.		E
AR_RA_DM_DISALLOWED	Revenue cannot be adjusted on debit memos or debit memo reversals		E
AR_RA_FULL_CREDIT	One or more credit memos have been applied for the full amount of this invoice		E
AR_RA_GL_DATE_CHANGED	GL date, &GL_DATE, is not in an open or future-enterable period. GL date has been changed to &NEW_GL_DATE		W
AR_RA_GUAR_DISALLOWED	Revenue cannot be adjusted on guarantees.		E
AR_RA_INVALID_AMOUNT_MODE	Amount mode &AMOUNT_MODE is invalid.		E
AR_RA_INVALID_CAT_SEGMENTS	This combination of category segments is invalid: &CONCAT_SEGS.		E
AR_RA_INVALID_CATEGORY	A valid category to which items belong that are currently on one or more lines on this transaction must be entered		E
AR_RA_INVALID_CATEGORY_ID	Category ID &CATEGORY_ID is invalid.		E

MESSAGE CODE	MESSAGE TEXT	DESCRIPTION	TYPE
AR_RA_INVALID_CODE_COMB	An error occurred while generating the following accounting flexfield code combination: &CODE_COMBINATION	This message is generated by the revenue adjustment API because of an error with the specified accounting flexfield code combination. Possible causes: segment values could not be found by AutoAccounting or have been disabled.	E
AR_RA_INVALID_ITEM	A valid item that is currently on one or more lines on this transaction must be entered		E
AR_RA_INVALID_ITEM_ID	Inventory item ID &ITEM_ID is invalid.		E
AR_RA_INVALID_ITEM_SEGMENTS	This combination of item segments is invalid: &CONCAT_SEGS.		E
AR_RA_INVALID_LINE_ID	Transaction line ID &CUST_TRX_LINE_ID is invalid.		E
AR_RA_INVALID_LINE_MODE	Line selection mode &LINE_MODE is invalid.		E
AR_RA_INVALID_REASON	Reason code &REASON_CODE is not a valid lookup code.		E
AR_RA_INVALID_SALESCRED_TYPE	Sales credit type &SALESCRED_TYPE is invalid.		E
AR_RA_INVALID_SALESREP_NUMBER	Salesperson number &SALESREP_NUMBER is invalid.		E
AR_RA_ITEM_NOT_ON_TRX	There are no lines with item &ITEM_ID on this transaction.		E
AR_RA_LINE_NOT_ON_TRX	There are no lines with line number &LINE_NUMBER on this transaction.		E

MESSAGE CODE	MESSAGE TEXT	DESCRIPTION	TYPE
AR_RA_NO_EARNED_REVENU E	There is no earned revenue on this transaction	This message is generated by the revenue adjustment API when there is no earned revenue on the selected transaction lines.	E
AR_RA_NO_FROM_CATEGORY	Please provide a from-category.		E
AR_RA_NO_FROM_ITEM	Please provide a from-item.		E
AR_RA_NO_FROM_LINE	Please provide a from-line.		E
AR_RA_NO_OPEN_PERIODS	The transaction date must fall during an open period or prior to a future period	This message is generated by the revenue adjustment API because there are no open or future periods relating to the transaction date or following the transaction date. Revenue cannot be posted to periods prior to the transaction date.	E
AR_RA_NO_REV_SALES_CREDI T	Line &LINE_NUMBER has no revenue sales credits	This message is generated by the revenue adjustment API when a transaction line with no sales credits is encountered.	E
AR_RA_NO_REV_TO_ADJUST	There is no adjustable revenue on the selected lines	This message is generated by the revenue adjustment API when there is no adjustable revenue on the selected transaction lines.	E
AR_RA_NO_SELECTED_SALESC RED	There are no sales credits for this line selection available to transfer		E
AR_RA_NO_TO_SALESREP	Please provide a valid salesperson number or ID for sales credit transfers or additions.		E
AR_RA_NO_TRX_NUMBER	Please provide a valid transaction number or ID.		E

MESSAGE CODE	MESSAGE TEXT	DESCRIPTION	TYPE
AR_RA_NO_UNEARNED_REVENUE	There is no unearned revenue on this transaction	This message is generated by the revenue adjustment API when there is no unearned revenue on the selected transaction lines.	E
AR_RA_PARTIAL_CREDIT	One or more partial credit memos have been applied against this invoice		W
AR_RA_PCT_EXCEEDS_AVAIL_PCT	The percentage entered is greater than &TOT_AVAIL_PCT, the total available percentage of adjustable revenue on the lines selected	This message is generated by the revenue adjustment API when there is insufficient adjustable revenue on the selected transaction lines to meet the specified percentage.	E
AR_RA_SALES_CREDIT_LIMIT	Revenue and non-revenue sales credits exceed &SALES_CREDIT_LIMIT percent for salesperson &SALESREP_NAME on line &LINE_NUMBER	This message is generated by the revenue adjustment API when the total percentage of revenue and non-revenue sales credits per salesperson per line exceeds the limit specified in system options.	E
AR_RA_SALESREP_NOT_ON_TRANSACTION	Salesperson &SALESREP_NAME does not have any sales credits on this transaction.		E
AR_RA_TRX_NOTFOUND	Transaction number &TRX_NUMBER cannot be found.		E
AR_RA_TRX_TOO_MANY_ROWS	There is more than one transaction with the transaction number &TRX_NUMBER. Please also provide a batch source to ensure uniqueness of the transaction.		E
AR_RA_ZERO_AMOUNT	Amount entered cannot be zero	This message is generated by the revenue accounting API when attempting to adjust an amount of zero.	E

MESSAGE CODE	MESSAGE TEXT	DESCRIPTION	TYPE
AR_RAPI_DESC_FLEX_INVALID	The entered values for the descriptive flexfield &DFF_NAME is invalid.		E
AR_TW_INCORRECT_SALESCR_EDIT	Revenue sales credit not equal to line amount or 100% for line &LINE_NUMBER.		E
AR_TAPI_TRANS_NOT_EXIST	Transaction does not exist. (CUSTOMER_TRX_ID: &CUSTOMER_TRX_ID).		E
AR_TAPI_INVALID_SALESREP_I_D	Invalid salesrep id. (SALESREP_ID: &SALESREP_ID)		E

A

Predefined Setup for Oracle Subledger Accounting

Data that Oracle Receivables Predefines for Oracle Subledger Accounting

Oracle Receivables provides predefined data for Oracle Subledger Accounting that you can use to integrate the two applications. When you run the Create Accounting program to create draft or final subledger accounting, the program uses the predefined data to determine how to create the accounting. Receivables predefines setup for Subledger Accounting so that the Create Accounting program accepts the default accounting information from AutoAccounting without change. Subledger Accounting transfers the final accounting to Oracle General Ledger.

You can optionally define your own subledger accounting rules to overwrite the default accounts from the accounting events.

Note: You must use an Oracle Receivables responsibility to query predefined data that is associated with the *Receivables* application.

The following sections describe the data that Receivables predefines in Subledger Accounting:

- Applications, page A-2
- Event Entities, page A-2
- Event Classes and Event Types, page A-3
- Process Categories, page A-4
- Accounting Event Class Options, page A-4
- Sources, Source Assignments, and Accounting Attribute Assignments, page A-5

- Journal Line Types, page A-6
- Account Derivation Rules, page A-12
- Journal Lines Definitions, page A-13
- Application Accounting Definitions, page A-15
- Subledger Accounting Methods, page A-17

Applications

Oracle Receivables predefines one application in Oracle Subledger Accounting named *Receivables*. Most of the data that Oracle Receivables predefines for Oracle Subledger Accounting is associated with the *Receivables* application.

The following table shows the attribute values that Oracle Receivables predefines for the *Receivables* application. The first column lists the fields and the second column lists the values for each field.

Predefined Receivables Application

Field	Value
Application Name	Receivables
Drilldown Procedure	AR_DRILLDOWN_PUB_PKG.DRILLDOWN
Use Security	Yes
Policy Function	XLA_SECURITY_POLICY_PKG.MO_POLICY
Journal Source	Receivables
Third Party Control Account Type	Customer
Subject to Validation	No
Calculate Reporting Currency Amounts	Yes

Event Entities

The following table lists the setup information that Oracle Receivables predefines for the event entities.

Predefined Event Entities

Application	Entity Name	Description	Gapless Event Processing
Receivables	Adjustments	Adjustments	No
Receivables	Bills Receivable	Bills Receivable	No
Receivables	Receipts	Receipts	No
Receivables	Transactions	Transactions	No

Event Classes and Event Types

Oracle Receivables predefines event classes and event types for each event entity that belongs to the *Receivables* application.

The following table lists the event classes and event types that Oracle Receivables predefines for the *Receivables* application.

Predefined Event Classes and Event Types for the Receivables Application

Entity	Event Class Name
Adjustments	Adjustment
Bills Receivable	Bills Receivable
Receipts	Miscellaneous Receipt
Receipts	Receipt
Transactions	Chargeback
Transactions	Credit Memo
Transactions	Debit Memo
Transactions	Deposit

Entity	Event Class Name
Transactions	Guarantee
Transactions	Invoice

Process Categories

Oracle Receivables predefines the following process categories:

- Adjustments
- Bills Receivable
- Miscellaneous Receipts
- Standard Receipts
- Third Party Merge
- Transactions

Accounting Event Class Options

Accounting event class options define attributes of an event class. Oracle Receivables defines the accounting event class options for each predefined event class.

The following table lists the accounting event class options that Oracle Receivables predefines for the *Receivables* application.

Predefined Accounting Event Class Options for the Receivables Application

Event Class	Process Category	Default Journal Category	Transaction View	Balance Types
Adjustment	Adjustments	Adjustment	AR_ADJ_INF_V	Actual
Bills Receivable	Bills Receivable	Bills Receivable	AR_TRX_INF_V	Actual
Miscellaneous Receipt	Miscellaneous Receipts	Misc Receipts	AR_CR_INF_V	Actual

Event Class	Process Category	Default Journal Category	Transaction View	Balance Types
Receipt	Standard Receipts	Receipts	AR_CR_INF_V	Actual
Chargeback	Transactions	Chargebacks	AR_TRX_INF_V	Actual
Credit Memo	Transactions	Credit Memos	AR_TRX_INF_V	Actual
Debit Memo	Transactions	Debit Memos	AR_TRX_INF_V	Actual
Deposit	Transactions	Sales Invoices	AR_TRX_INF_V	Actual
Guarantee	Transactions	Sales Invoices	AR_TRX_INF_V	Actual
Invoice	Transactions	Sales Invoices	AR_TRX_INF_V	Actual

Sources, Source Assignments, and Accounting Attribute Assignments

Oracle Receivables predefines sources, source assignments, and accounting attribute assignments for Oracle Subledger Accounting.

You can use the Accounting Methods Builder to review the sources, source assignments, and accounting attribute assignments. You must access the Accounting Methods Builder using an Oracle Receivables responsibility if you want to review the sources, source assignments, and accounting attribute assignments associated with the *Receivables* application.

Note: You cannot make changes to predefined sources, source assignments, or accounting attribute assignments. However, you can define your own custom sources.

If you choose to define your own journal line types or application accounting definitions, then you can override the default accounting attribute assignments.

Oracle Receivables provides numerous predefined sources. When you use the Sources window to review the predefined sources, you can optionally export the queried sources from the application to a Microsoft Excel spreadsheet.

To export a list of sources:

1. From an Oracle Receivables responsibility, navigate to the Sources window.
2. Query the records you want to export.

3. Place your cursor in the multi-row block that contains the records to be exported.
4. Choose Export from the File menu.

See: Exporting Records to a File, *Oracle Applications User's Guide*.

Journal Line Types

Oracle Receivables predefines journal line types for each predefined event class. Oracle Receivables specifies conditions for the use of each journal line type.

The following table lists the journal line types that Oracle Receivables predefines for the *Receivables* application.

Predefined Journal Line Types for the Receivables Application

Event Class	Name	Balance Type	Side
Adjustment	Adjustment	Actual	Credit
Adjustment	Adjustment Bills Receivable Endorsement	Actual	Credit
Adjustment	Adjustment Charge	Actual	Credit
Adjustment	Adjustment Default Receivable	Actual	Credit
Adjustment	Adjustment Deferred Tax	Actual	Credit
Adjustment	Adjustment Charge Non-Recoverable Tax	Actual	Credit
Adjustment	Adjustment Non-Recoverable Tax	Actual	Credit
Adjustment	Adjustment Tax	Actual	Credit
Bills Receivable	Bills Receivable Deferred Tax	Actual	Credit
Bills Receivable	Bills Receivable Factoring	Actual	Credit
Bills Receivable	Bills Receivable Receivable	Actual	Credit

Event Class	Name	Balance Type	Side
Bills Receivable	Bills Receivable Remittance	Actual	Credit
Bills Receivable	Bills Receivable Tax	Actual	Credit
Bills Receivable	Bills Receivable Unpaid	Actual	Credit
Chargeback	Chargeback Default Receivable	Actual	Debit
Chargeback	Chargeback Revenue	Actual	Credit
Credit Memo	Credit Memo Default Deferred Tax Application	Actual	Credit
Credit Memo	Credit Memo Default Application	Actual	Credit
Credit Memo	Credit Memo Default Tax Application	Actual	Credit
Credit Memo	Credit Memo Refund Application	Actual	Credit
Credit Memo	Credit Memo Charges	Actual	Credit
Credit Memo	Credit Memo Default Receivable	Actual	Debit
Credit Memo	Credit Memo Default Revenue	Actual	Credit
Credit Memo	Credit Memo Default Tax	Actual	Credit
Credit Memo	Credit Memo Rounding	Actual	Credit
Credit Memo	Credit Memo Unbilled Receivable	Actual	Credit
Credit Memo	Credit Memo Deferred Revenue	Actual	Credit

Event Class	Name	Balance Type	Side
Debit Memo	Debit Memo Charges	Actual	Credit
Debit Memo	Debit Memo Default Receivable	Actual	Debit
Debit Memo	Debit Memo Freight	Actual	Credit
Debit Memo	Debit Memo Revenue	Actual	Credit
Debit Memo	Debit Memo Rounding	Actual	Credit
Debit Memo	Debit Memo Tax	Actual	Credit
Debit Memo	Debit Memo Unbilled Receivable	Actual	Debit
Debit Memo	Debit Memo Deferred Revenue	Actual	Credit
Deposit	Deposit Default Receivable	Actual	Debit
Deposit	Deposit Offset	Actual	Credit
Guarantee	Guarantee Default Receivable	Actual	Debit
Guarantee	Guarantee Revenue	Actual	Credit
Invoice	Invoice Charges	Actual	Credit
Invoice	Invoice Default Receivable	Actual	Debit
Invoice	Invoice Freight	Actual	Credit
Invoice	Invoice Revenue	Actual	Credit
Invoice	Invoice Rounding	Actual	Credit
Invoice	Invoice Tax	Actual	Credit
Invoice	Invoice Unbilled Receivable	Actual	Credit

Event Class	Name	Balance Type	Side
Invoice	Invoice Deferred Revenue	Actual	Credit
Miscellaneous Receipt	Miscellaneous Receipt Bank Charges	Actual	Credit
Miscellaneous Receipt	Miscellaneous Receipt Cleared Cash	Actual	Credit
Miscellaneous Receipt	Miscellaneous Receipt Confirmed Cash	Actual	Credit
Miscellaneous Receipt	Miscellaneous Receipt Short Term Debt	Actual	Credit
Miscellaneous Receipt	Miscellaneous Receipt Factored Cash	Actual	Credit
Miscellaneous Receipt	Miscellaneous Receipt Miscellaneous Cash	Actual	Credit
Miscellaneous Receipt	Miscellaneous Receipt Remitted Cash	Actual	Credit
Miscellaneous Receipt	Miscellaneous Receipt Tax	Actual	Credit
Receipt	Receipt On Account Application	Actual	Credit
Receipt	Receipt Application to Freight	Actual	Credit
Receipt	Receipt Application to Revenue	Actual	Credit
Receipt	Receipt Application to Rounding	Actual	Credit
Receipt	Receipt Application to Suspense Revenue	Actual	Credit
Receipt	Receipt Application to Tax	Actual	Credit

Event Class	Name	Balance Type	Side
Receipt	Receipt Application to Unbilled Revenue	Actual	Credit
Receipt	Receipt Application to Earned Revenue	Actual	Credit
Receipt	Receipt Bank Charges	Actual	Credit
Receipt	Receipt Cleared Cash	Actual	Credit
Receipt	Receipt Credit Card Chargeback Application	Actual	Credit
Receipt	Receipt Chargeback Application	Actual	Credit
Receipt	Receipt Claim Investigation Application	Actual	Credit
Receipt	Receipt Confirmed Cash	Actual	Credit
Receipt	Receipt Currency Rounding	Actual	Credit
Receipt	Receipt Short Term Debt	Actual	Credit
Receipt	Receipt Default Application	Actual	Credit
Receipt	Receipt Deferred Tax	Actual	Credit
Receipt	Receipt Earned Discount	Actual	Credit
Receipt	Receipt Earned Discount on Freight	Actual	Debit
Receipt	Receipt Earned Discount Non-Recoverable Tax	Actual	Credit
Receipt	Receipt Earned Discount on Revenue	Actual	Debit

Event Class	Name	Balance Type	Side
Receipt	Receipt Earned Discount on Tax	Actual	Debit
Receipt	Receipt Exchange Gain	Actual	Credit
Receipt	Receipt Exchange Gain Loss	Actual	Gain/Loss
Receipt	Receipt Exchange Loss	Actual	Credit
Receipt	Receipt Factored Cash	Actual	Credit
Receipt	Receipt Factored Bills Receivable	Actual	Credit
Receipt	Receipt Payment Netting Application	Actual	Credit
Receipt	Receipt Prepayment Application	Actual	Credit
Receipt	Receipt Refund Application	Actual	Credit
Receipt	Receipt Remitted Cash	Actual	Credit
Receipt	Receipt Remitted Bills Receivable	Actual	Credit
Receipt	Receipt Tax	Actual	Credit
Receipt	Receipt Unapplied Cash	Actual	Debit
Receipt	Unapplied Cash	Actual	Debit
Receipt	Receipt Unapplied for Gain Loss lines	Actual	Debit
Receipt	Receipt Unearned Discount	Actual	Credit
Receipt	Receipt Unearned Discount on Freight	Actual	Debit

Event Class	Name	Balance Type	Side
Receipt	Receipt Unearned Discount on Non Recoverable Tax	Actual	Credit
Receipt	Receipt Unearned Discount on Revenue	Actual	Debit
Receipt	Receipt Unearned Discount on Tax	Actual	Debit
Receipt	Receipt Unidentified Cash	Actual	Credit
Receipt	Receipt Write-Off Application	Actual	Credit

Account Derivation Rules

Oracle Receivables predefines account derivation rules. When Oracle Subledger Accounting uses the predefined account derivation rules that Oracle Receivables provides, it accepts the default accounting that Oracle Receivables generates using AutoAccounting without change.

You can optionally define your own account derivation rules for an Accounting Flexfield or for a segment. In this case, Oracle Subledger Accounting overrides the default accounts that Oracle Receivables generates, or individual segment values in the default accounts, when it creates the draft or final subledger accounting.

The name account derivation rules that Oracle Receivables predefines for the *Receivables* application are as follows:

- Credit Memo Distribution GL Account
- Collection Bank Charges Account
- Bills Under Collection Account
- Bills Under Discount Account
- Collection Bank Account Cash Account
- Collection Endorsement Account
- Discount Endorsement Account
- Collection Bank Factoring Charges Account

- Distribution GL Account
- Remit Bank Unapplied account
- System Gain GL Account
- System Loss GL Account
- Transaction Distribution GL Account
- Transaction Distribution GL Account with reference

Journal Lines Definitions

Oracle Receivables predefines journal lines definitions that group the predefined journal line types and account derivation rules within each of the predefined event types. Oracle Receivables assigns each predefined journal lines definition to all event types within an event class.

The following table lists the journal lines definitions that Oracle Receivables predefines for the *Receivables* application.

Predefined Journal Lines Definitions for the Receivables Application

Event Class	Journal Lines Definition Name	Journal Line Types
Adjustment	Adjustments - Default Accrual	Adjustment, Adjustment Bills Receivable Endorsement, Adjustment Charge, Adjustment Charge Non-Recoverable Tax, Adjustment Default Receivable, Adjustment Deferred Tax, Adjustment Non-Recoverable Tax, Adjustment Tax
Bills Receivable	Receivables Bills Journal Lines Definition	Bills Receivable Deferred Tax, Bills Receivable Factoring, Bills Receivable Receivable, Bills Receivable Remittance, Bills Receivable Tax, Bills Receivable Unpaid
Chargeback	Chargebacks - Default Accrual	Chargeback Default Receivable, Chargeback Revenue
Credit Memo	Credit Memos - Default Accrual	Credit Memo Charges, Credit Memo Default Application, Credit Memo Default Deferred Tax Application, Credit Memo Default Receivable, Credit Memo Default Revenue, Credit Memo Default Tax, Credit Memo Default Tax Application, Credit Memo Deferred Revenue, Credit Memo Refund Application, Credit Memo Rounding, Credit Memo Unbilled Receivable

Event Class	Journal Lines Definition Name	Journal Line Types
Debit Memo	Debit Memos - Default Accrual	Debit Memo Charges, Debit Memo Default Receivable, Debit Memo Deferred Revenue, Debit Memo Freight, Debit Memo Revenue, Debit Memo Rounding, Debit Memo Tax, Debit Memo Unbilled Receivable
Deposit	Deposits - Default Accrual	Deposit Default Receivable, Deposit Offset
Guarantee	Guarantees - Default Accrual	Guarantee Default Receivable, Guarantee Revenue
Invoice	Invoices - Default Accrual	Invoice Charges, Invoice Default Receivable, Invoice Deferred Revenue, Invoice Freight, Invoice Revenue, Invoice Rounding, Invoice Tax, Invoice Unbilled Receivable
Miscellaneous Receipt	Miscellaneous Receipts	Miscellaneous Receipt Bank Charges, Miscellaneous Receipt Cleared Cash, Miscellaneous Receipt Confirmed Cash, Miscellaneous Receipt Factored Cash, Miscellaneous Receipt Miscellaneous Cash, Miscellaneous Receipt Remitted Cash, Miscellaneous Receipt Short Term Debt, Miscellaneous Receipt Tax
Receipt	Receipts - Default Accrual	Receipt Bank Charges, Receipt Chargeback Application, Receipt Claim Investigation Application, Receipt Cleared Cash, Receipt Confirmed Cash, Receipt Credit Card Chargeback Application, Receipt Currency Rounding, Receipt Default Application, Receipt Deferred Tax, Receipt Earned Discount, Receipt Earned Discount Non-Recoverable Tax, Receipt Exchange Gain Loss, Receipt Factored Bills Receivable, Receipt Factored Cash, Receipt On Account Application, Receipt Payment Netting Application, Receipt Prepayment Application, Receipt Refund Application, Receipt Remitted Bills Receivable, Receipt Remitted Cash, Receipt Short Term Debt, Receipt Tax, Receipt Unapplied Cash, Receipt Unearned Discount, Receipt Unearned Discount on Non Recoverable Tax, Receipt Unidentified Cash, Receipt Write-Off Application, Unapplied Cash

Event Class	Journal Lines Definition Name	Journal Line Types
Receipt	Receipt - Basis Journal Lines Definition	Receipt Application to Earned Revenue, Receipt Application to Freight, Receipt Application to Revenue, Receipt Application to Rounding, Receipt Application to Suspense Revenue, Receipt Application to Tax, Receipt Application to Unbilled Revenue, Receipt Bank Charges, Receipt Claim Investigation Application, Receipt Cleared Cash, Receipt Confirmed Cash, Receipt Currency Rounding, Receipt Earned Discount, Receipt Earned Discount on Freight, Receipt Earned Discount on Revenue, Receipt Earned Discount on Tax, Receipt Exchange Gain Loss, Receipt Factored Bills Receivable, Receipt Factored Cash, Receipt On Account Application, Receipt Payment Netting Application, Receipt Prepayment Application, Receipt Refund Application, Receipt Remitted Bills Receivable, Receipt Remitted Cash, Receipt Short Term Debt, Receipt Unapplied Cash, Receipt Unapplied for Gain Loss lines, Receipt Unearned Discount, Receipt Unearned Discount on Freight, Receipt Unearned Discount on Revenue, Receipt Unearned Discount on Tax, Receipt Unidentified Cash, Receipt Write-Off Application, Unapplied Cash

Application Accounting Definitions

Oracle Receivables predefines the *Receivables Default Accrual* and the *Receivables Default Cash Basis Accounting Definition* application accounting definitions.

The following table lists the assignments for the *Receivables Default Accrual* application accounting definition that Oracle Receivables predefines for the *Receivables* application.

Assignments for the Predefined Receivables Default Accrual Application Accounting Definition

Event Class Assignments	Event Type Assignments	Create Accounting	Journal Line Definition Assignments
Adjustment	All	Yes	Adjustments - Default Accrual
Bills Receivable	All	Yes	Receivables Bills Journal Lines Definition
Chargeback	All	Yes	Chargebacks - Default Accrual

Event Class Assignments	Event Type Assignments	Create Accounting	Journal Line Definition Assignments
Credit Memo	All	Yes	Credit Memos - Default Accrual
Debit Memo	All	Yes	Debit Memos - Default Accrual
Deposit	All	Yes	Deposits - Default Accrual
Guarantee	All	Yes	Guarantees - Default Accrual
Invoice	All	Yes	Invoices - Default Accrual
Miscellaneous Receipt	All	Yes	Miscellaneous Receipts
Receipt	All	Yes	Receipts - Default Accrual

The following table lists the assignments for the *Receivables Default Cash Basis Accounting Definition* application accounting definition that Oracle Receivables predefines for the *Receivables* application.

Assignments for the Predefined Receivables Default Cash Basis Accounting Definition Application Accounting Definition

Event Class Assignments	Event Type Assignments	Create Accounting	Journal Line Definition Assignments
Adjustment	All	No	
Bills Receivable	All	No	
Chargeback	All	No	
Credit Memo	All	No	
Debit Memo	All	No	
Deposit	All	No	
Guarantee	All	No	

Event Class Assignments	Event Type Assignments	Create Accounting	Journal Line Definition Assignments
Invoice	All	No	
Miscellaneous Receipt	All	Yes	Miscellaneous Receipts
Receipt	All	Yes	Receipt - Basis Journal Lines Definition

Subledger Accounting Methods

Oracle Subledger Accounting provides predefined subledger accounting methods that group the predefined application accounting definitions for subledger applications. You can optionally create your own subledger accounting methods.

Oracle Receivables assigns the predefined *Receivables Default Accrual* application accounting definition to the predefined *Standard Accrual* subledger accounting method, and the *Receivables Default Cash Basis Accounting Definition* application accounting definition to the predefined *Standard Cash* subledger accounting method.

You can assign these subledger accounting methods to your ledgers.

B

Oracle Receivables Table and Column Descriptions

AutoInvoice Table and Column Descriptions

Below is a detailed description of the four interface tables Receivables uses to temporarily store transaction data from your original system. Each column has important, detailed information you need to know to successfully run AutoInvoice. AutoInvoice uses the fifth table, RA_INTERFACE_ERRORS_ALL, to store information about interface data that failed validation.

Interface Tables

- RA_INTERFACE_LINES_ALL, page B-1
- RA_INTERFACE_SALESCREDITS_ALL, page B-49
- RA_INTERFACE_DISTRIBUTIONS_ALL, page B-53
- AR_INTERFACE_CONTS_ALL, page B-58
- RA_INTERFACE_ERRORS_ALL, page B-60

Table Name: RA_INTERFACE_LINES_ALL

This table stores transaction header and line information. AutoInvoice uses Transaction Flexfields to uniquely identify each transaction that you import into Receivables. AutoInvoice always uses the Line Transaction Flexfield structure for both the Link-to and Reference information when importing invoices. For more information, see: Transaction Flexfields, *Oracle Receivables User Guide*.

Important: The following columns of the AutoInvoice interface table RA_INTERFACE_LINES_ALL have become obsolete in R12:

- CUSTOMER_BANK_ACCOUNT_ID
- CUSTOMER_BANK_ACCOUNT_NAME
- PAYMENT_SERVER_ORDER_NUM
- APPROVAL_CODE

Instead of these fields, Autoinvoice requires you to populate the PAYMENT_TRXN_EXTENSION_ID. You can obtain this by calling the IBY_FNDCPT_TRXN_PUB.Create_Transaction_Extension before populating the Autoinvoice tables.

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_ALL.ACOUNTING_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_ALL.ACOUNTING_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 if 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. 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 neither 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.

If this line has AMOUNT_INCLUDES_TAX set to Yes, the sales credits and line amounts for this column must include tax.

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 Create Clearing is set to No for this transaction batch source (suspense/clearing account not used), RA_CUSTOMER_TRX_LINES_ALL.REVENUE_AMOUNT and RA_CUSTOMER_TRX_LINES_ALL.EXTENDED_AMOUNT.
	If Create Clearing is set to Yes for this transaction batch source (suspense/clearing account used), RA_CUSTOMER_TRX_LINES_ALL.REVENUE_AMOUNT

AMOUNT_INCLUDES_TAX_FLAG

This column controls whether the amount for this transaction line includes tax. If this column is set to 'Y', this line is assigned to a tax inclusive tax code. If this is a tax group, this column should be null.

AutoInvoice only uses this column if the tax code assigned to this line has Allow Override set to Yes for the Tax Inclusive calculation.

Populate this column for invoices only. For regular credit memos, AutoInvoice always uses the AMOUNT_INCLUDES_TAX_FLAG value from the invoice that you are crediting.

Validation:	If this is a tax code and Allow Override is set to No, this should be equal to either the setting of the Amount Includes Tax option for this tax code or null. Additionally, if Allow Override is set to No the Amount Includes Tax flag at the line level must equal the Allow Override flag for this tax code.
Destination:	RA_CUSTOMER_TRX_LINES_ALL.AMOUNT_INCLUDES_TAX_FLAG

ADDRESS_VERIFICATION_CODE

The credit card address verification code provided by Oracle Payment Server.

Validation	None
Destination	RA_CUSTOMER_TRX_ALL.ADDRESS_VERIFICATION_CODE

ATTRIBUTE1-15

Enter the Invoice Line Information 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_ALL.ATTRIBUTE1-15

Note: To ensure that AutoInvoice accurately groups your imported invoices, do not include newline or carriage return characters (chr(10) or chr(13)) in these Descriptive Flexfield columns.

ATTRIBUTE_CATEGORY

Enter the Invoice Line Information 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_ALL.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_ALL.NAME and RA_BATCH_SOURCES_ALL.BATCH_SOURCE_TYPE = 'FOREIGN'

Destination: RA_BATCHES_ALL.BATCH_SOURCE_ID and RA_CUSTOMER_TRX_ALL.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 you are passing freight for a specific line, do not enter text in this column.

Validation:	None
Destination:	RA_CUSTOMER_TRX_ALL.COMMENTS

CONS_BILLING_NUMBER

Enter the number for this consolidated bill.

A consolidated bill number is used for grouping a set of invoices under one bill.

Validation:	Must not already exist in AR_CONS_INV_ALL.CONS_BILLING_NUMBER and AR_CONS_INV_ALL.CONS_INV_TYPE='MINV'
Destination:	AR_CONS_INV_ALL.CONS_BILLING_NUMBER

CONVERSION_DATE

Enter the exchange rate date for this transaction. If you do not enter a date, AutoInvoice uses the transaction date as the default. If the currency of the transaction line is the same as the base currency, then leave this column null. If a credit memo is being processed, AutoInvoice uses the conversion date of the invoice that the credit memo is against, and not the credit memo transaction date.

Validation:	None
Destination:	RA_CUSTOMER_TRX_ALL.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.CONVERSION_TYPE = 'User' then this column must not be null; otherwise, it must be null.
Destination:	RA_CUSTOMER_TRX_ALL.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_ALL.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 this transaction is a credit memo against a transaction which uses an accounting rule and CREDIT_METHOD_FOR_ACCT_RULE is UNIT, then AutoInvoice rejects the credit memo if the credit quantity exceeds the quantity on the target invoice line.

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_ALL.CREDIT_METHOD_FOR_RULES

CREDIT_METHOD_FOR_INSTALLMENTS

Enter the credit method for crediting a transaction that uses split payment terms. Choices include PRORATE, LIFO, or FIFO.

If this transaction is a credit memo against a transaction that 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_ALL.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_ALL.INVOICE_CURRENCY_CODE and AR_PAYMENT_SCHEDULES_ALL.INVOICE_CURRENCY_CODE

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_ALL.CUSTOMER_TRX_ID, AR_PAYMENT_SCHEDULES_ALL.CUSTOMER_TRX_ID, RA_CUSTOMER_TRX_LINES_ALL.CUSTOMER_TRX_ID, and RA_CUST_TRX_LINE_GL_DIST_ALL.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_ALL.CUST_TRX_TYPE_ID
Destination:	RA_CUSTOMER_TRX_ALL.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_ALL.NAME

Destination: None

DESCRIPTION

This is a required column in AutoInvoice. Enter the description for this transaction.

Validation: None

Destination: RA_CUSTOMER_TRX_LINES_ALL.DESCRIPTION

DOCUMENT_NUMBER

Enter the document number for this transaction.

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.

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.

If LINE_TYPE = 'TAX' or you are passing freight for a specific line, do not enter a value in this column.

Validation: Number must not already exist in Oracle Receivables

Destination: RA_CUSTOMER_TRX_ALL.DOC_SEQUENCE_VALUE

DOCUMENT_NUMBER_SEQUENCE_ID

This column is used by AutoInvoice and should be left null. AutoInvoice uses this column to store the document sequence ID for this transaction.

Validation: None

Destination: RA_CUSTOMER_TRX_ALL.DOC_SEQUENCE_ID

EXCEPTION_ID

This column is used by AutoInvoice and should be left null. AutoInvoice defaults a value in this column when a tax exception occurs.

If your transaction is a credit memo, AutoInvoice defaults the tax exemption ID of the transaction you are crediting.

Validation: None

Destination: RA_CUSTOMER_TRX_LINES_ALL.ITEM_EXCEPTION_RATE_ID

EXEMPTION_ID

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.

For credit memos AutoInvoice defaults the tax exception ID of the transaction you are crediting.

Validation: None

Destination: RA_CUSTOMER_TRX_LINES_ALL.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_ALL.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.

For a more details on general ledger dates, see: Determining Dates, *Oracle Receivables User Guide*.

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.

If your invoice uses Bill in Arrears as the invoicing rule, then the GL_DATE column should be NULL.

Destination: RA_CUST_TRX_LINE_GL_DIST_ALL.GL_DATE

HEADER_ATTRIBUTE1-15

Enter Descriptive Flexfield attribute information for the Transaction Information Flexfield. Descriptive Flexfield attributes let you 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_ALL.ATTRIBUTE1-15

Note: To ensure that AutoInvoice accurately groups your imported invoices, do not include newline or carriage return characters (chr(10) or chr(13)) in these Descriptive Flexfield columns.

HEADER_ATTRIBUTE_CATEGORY

For the Transaction Information Flexfield, 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_ALL.ATTRIBUTE_CATEGORY

HEADER_GDF_ATTRIBUTE1-30

Reserved for country-specific functionality.

Validation: Performed by Oracle Global Financials

Destination: RA_CUSTOMER_TRX_ALL.GLOBAL_ATTRIBUTE1-30

HEADER_GDF_ATTR_CATEGORY

Reserved for country-specific functionality.

Validation: Performed by Oracle Global Financials

Destination: RA_CUSTOMER_TRX_ALL.GLOBAL_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_ALL.INITIAL_CUSTOMER_TRX_ID

INTERFACE_LINE_ATTRIBUTE1-15

Enter the Line Transaction Flexfield for this transaction. The Line Transaction Flexfield is a combination of attribute values that you use to uniquely identify this transaction line in your original system. The reference value you enter here provides you with an audit trail from Receivables back to your original system. You must enter values for enabled attributes.

Receivables copies the Line Transaction Flexfield to the Invoice Transaction Flexfield. When you import transactions with multiple lines using AutoInvoice, the attributes of the first line from the ordered lines will appear in the Invoice Transaction Flexfield.

Note: Interface lines belonging to the same transaction are ordered by the following SQL clause:

```
waybill_number || ship_via asc,  
ship_date_actual desc
```

If a transaction has only one line, then the Invoice Transaction Flexfield will be the same as the Line Transaction Flexfield.

Validation:	Must not already exist together with INTERFACE_LINE_CONTEXT in RA_INTERFACE_LINES_ALL and RA_CUSTOMER_TRX_LINES_ALL.
	All enabled attributes for a given INTERFACE_LINE_CONTEXT must have values. Different attribute columns may be enabled depending on the value in the INTERFACE_LINE_CONTEXT column.
Destination:	RA_CUSTOMER_TRX_ALLINTERFACE_HEADER_ATT RIBUTE1-15 and RA_CUSTOMER_TRX_LINES_ALLINTERFACE_LINE_A TTRIBUTE1-15
	Note: To ensure that AutoInvoice accurately groups your imported invoices, do not include newline or carriage return characters (chr(10) or chr(13)) in these Descriptive Flexfield columns.

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 information with global context, set this column to 'Global Data Elements'.

Validation:	None
Destination:	RA_CUSTOMER_TRX_ALLINTERFACE_HEADER_CONTEXT and RA_CUSTOMER_TRX_LINES_ALLINTERFACE_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_ALLCUSTOMER_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_ALL.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_ALL.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 you must also enter an accounting rule. If LINE_TYPE = 'CHARGES' then this column must be null.
Destination:	RA_CUSTOMER_TRX_ALL.INVOICING_RULE_ID

INVOICING_RULE_NAME

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 INVOICING_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 (inclusive).
Destination:	RA_CUSTOMER_TRX_LINES_ALL.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_GDF_ATTRIBUTE1-20

Reserved for country-specific functionality.

Validation: Performed by Oracle Global Financials

Destination: RA_CUSTOMER_TRX_LINES_ALL.GLOBAL_ATTRIBUTE1-20

LINE_GDF_ATTR_CATEGORY

Reserved for country-specific functionality.

Validation: Performed by Oracle Global Financials

Destination: RA_CUSTOMER_TRX_LINES_ALL.GLOBAL_ATTRIBUTE_CATEGORY

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. (CHARGES refers to finance charges.) 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_ALL.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 with a line that has 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: None

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_ALL.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_ALL.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_ALL.MEMO_LINE_ID

Destination: RA_CUSTOMER_TRX_LINES_ALL.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_ALL.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_ALL.MOVEMENT_ID with
RA_INTERFACE_LINES_ALL.MOVEMENT_ID and updates

MTL_MOVEMENT_STATISTICS with transaction information (for example, customer_trx_id, batch_id, customer_trx_line_id).

Validation: None

Destination: RA_CUSTOMER_TRX_LINES_ALL.MOVEMENT_ID

MTL_SYSTEM_ITEMS_SEG1-20

Assign a System Item Flexfield value for each segment you enable in Receivables. 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

Enter the ID of the organization that this transaction belongs to. This column is mandatory in a multiple organization environment.

Validation: AutoInvoice imports transactions whose ORG_ID matches the value of the MO: Operating Unit profile option.

Destination: None.

ORIGINAL_GL_DATE

Stores the value of the GL_DATE column before AutoInvoice modifies the GL date. This column is used by AutoInvoice and should not be populated by the user.

Validation: None

Destination: None

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_ALL.ORIG_SYSTEM_BATCH_NA
ME

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, then AutoInvoice uses the Bill-To address to determine the Remit-To address for the customer. If the Remit-To address cannot be determined, then AutoInvoice will reject the 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_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 = HZ_CUST_ACCT_SITE.CUSTOMER_SITE_ID and
RA_INTERFACE_LINES_ALL.ORIG_SYSTEM_BILL_CUSTOMER_ID =
HZ_CUST_ACCOUNTS.CUST_ACCOUNT_ID and
HZ_CUST_ACCOUNTS.CUST_ACCOUNT_ID =
HZ_CUST_ACCT_SITE.CUST_ACCOUNT_ID and
HZ_CUST_ACCT_SITE.CUSTOMER_SITE_ID =
HZ_CUST_SITEUSES.CUST_ACCT_SITE_ID and
RA_SITEUSES.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 Receivables back to your original system.

If no default Remit_To Address has been specified, then AutoInvoice uses the Bill-To

address to determine the Remit-To address for the customer. If the Remit-To address cannot be determined, then AutoInvoice will reject the 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_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 =
H2_PARTY_SITES.ORIG_SYSTEM_REFERENCE and
CUSTOMER_REF =
H2_CUST_ACCOUNTS.ORIG_SYSTEM_REFERENCE and
H2_CUST_ACCOUNTS.CUST_ACCOUNT_ID =
H2_CUST_ACCT_SITE.CUST_ACCOUNT_ID and
H2_CUST_ACCT_SITE.CUSTOMER_SITE_ID =
H2_CUST_SITEUSES.CUST_ACCT_SITE_ID and
RA_SITEUSES.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 =
H2_CUST_ACCT_ROLES.CUST_ACCOUNT_ID and
RA_INTERFACE_LINES_ALL.ORIG_SYSTEM_BILL_CONTACT_ID = H2_CUST_SITEUSES.CUSTOMER_SITE_ID

Destination: RA_CUSTOMER_TRX_ALL.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 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 =
HZ_CUST_ACCT_ROLES.CUST_ACCOUNT_ID and
RA_INTERFACE_LINES_ALL.ORIG_SYSTEM_BILL_CONTACT_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
HZ_CUST_ACCOUNTS.CUST_ACCOUNT_ID

Destination: RA_CUSTOMER_TRX_ALL.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 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
HZ_CUST_ACCOUNTS.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 = HZ_CUST_ACCT_SITE.CUSTOMER_SITE_ID and
RA_INTERFACE_LINES_ALL.ORIG_SYSTEM_SHIP_CUSTOMER_ID =
HZ_CUST_ACCOUNTS.CUST_ACCOUNT_ID and
HZ_CUST_ACCOUNTS.CUST_ACCOUNT_ID =
HZ_CUST_ACCT_SITE.CUST_ACCOUNT_ID and
HZ_CUST_ACCT_SITE.CUSTOMER_SITE_ID =
HZ_CUST_SITEUSES.CUST_ACCT_SITE_ID and
HZ_CUST_SITEUSES.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 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_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 =
HZ_PARTY_SITES.ORIG_SYSTEM_REFERENCE and
RA_INTERFACE_LINES_ALL.ORIG_SYSTEM_SHIP_CUSTOMER_ID =
HZ_CUST_ACCOUNTS.CUST_ACCOUNT_ID and
HZ_CUST_ACCOUNTS.CUST_ACCOUNT_ID =
HZ_CUST_ACCT_SITE.CUST_ACCOUNT_ID and
HZ_CUST_ACCT_SITE.CUSTOMER_SITE_ID =
HZ_CUST_SITEUSES.CUST_ACCT_SITE_ID and
HZ_CUST_SITEUSES.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_CUSTOMER_ID =
HZ_CUST_ACCT_ROLES.CUST_ACCOUNT_ID and
RA_INTERFACE_LINES_ALL.ORIG_SYSTEM_SHIP_CONTACT_ID = HZ_CUST_SITEUSES.CUSTOMER_SITE_ID

Destination: RA_CUSTOMER_TRX_ALL.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 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_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_CUSTOMER_ID =
HZ_CUST_ACCT_ROLES.CUST_ACCOUNT_ID and
RA_INTERFACE_LINES_ALL.ORIG_SYSTEM_SHIP_CONTACT_REF =
HZ_CUST_ACCOUNT_ROLES.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
HZ_CUST_ACCOUNTS.CUST_ACCOUNT_ID

Destination: RA_CUSTOMER_TRX_ALL.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 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_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
HZ_CUST_ACCOUNTS.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
HZ_CUST_ACCOUNTS.CUST_ACCOUNT_ID

Destination: RA_CUSTOMER_TRX_ALL.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 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 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
HZ_CUST_ACCOUNTS.ORIG_SYSTEM_REFERENCE

Destination: None

OVERRIDE_AUTO_ACCOUNTING_FLAG

This column controls whether the code combination ID of the Accounting Flexfield for this accounting distribution, populated by the feeder system, should override AutoAccounting.

Populate this column for invoices and credit memos.

Validation: Value should be Y or N.

Destination: RA_CUSTOMER_TRX_LINES_ALL.OVERRIDE_AUTO_ACCOUNTING_FLAG

PAYMENT_SET_ID

This column contains a unique internal ID number that matches prepaid invoices with their prepayment receipts. This column should be populated only within a prepayments flow.

Validation: Must exist in
AR_RECEIVABLE_APPLICATIONS_ALL.PAYMENT_SET_ID

Destination: RA_CUSTOMER_TRX_LINES_ALL.PAYMENT_SET_ID

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

Destination: RA_CUSTOMER_TRX_ALL.PAYING_CUSTOMER_ID

PAYING_SITE_USE_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 site use.

Validation: None

Destination: RA_CUSTOMER_TRX_ALL.PAYING_SITE_USE_ID

PREVIOUS_CUSTOMER_TRX_ID

This column is used by AutoInvoice and should be left null.

For credit memos, AutoInvoice defaults a value into this column using RA_INTERFACE_LINES_ALL.REFERENCE_LINE_ID.

Validation: None

Destination: RA_CUSTOMER_TRX_ALL.PREVIOUS_CUSTOMER_TRX_ID and RA_CUSTOMER_TRX_LINES_ALL.CUSTOMER_TRX_ID

PRIMARY_SALESREP_ID

Enter the primary salesperson ID for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, and you entered Yes for the Require Salesreps 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 ID 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: Must exist in RA_SALESREPS.SALESREP_ID

Destination: RA_CUSTOMER_TRX_ALL.PRIMARY_SALESREP_ID

PRIMARY_SALESREP_NUMBER

Enter the primary salesperson number for this transaction.

If LINE_TYPE = 'LINE', 'CHARGES', or you are passing header freight, and you entered Yes for the Require Salesreps 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_ALL.PRINTING_OPTION

PROMISED_COMMITMENT_AMOUNT

Enter the amount of an existing deposit to use as payment towards a specific transaction.

When an order is imported into Receivables with a value in this column, Receivables adjusts the resulting invoice and reduces the deposit balance by the lesser of the promised amount, the commitment balance, or the remaining amount due on the invoice.

If this column has no value, then the commitment adjustment will be for the lesser of the total outstanding commitment or the total balance of the transaction.

Validation: None

Destination: None

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_ALL.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_ALL.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_ALL.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.

If this is a Credit Memo line and LINE_TYPE = 'LINE',

CREDIT_METHOD_FOR_ACCT_RULE = 'UNIT' then this column is mandatory.

If this transaction is a credit memo against a transaction which uses an accounting rule and CREDIT_METHOD_FOR_ACCT_RULE is UNIT, then AutoInvoice rejects the credit memo if the credit quantity exceeds the quantity on the target invoice line.

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 not be null.

Destination: RA_CUSTOMER_TRX_LINES_ALL.QUANTITY_INVOICED if this transaction is an invoice line.
RA_CUSTOMER_TRX_LINES_ALL.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_ALL.QUANTITY_ORDERED

D

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_ALL.REASON_CODE and RA_CUSTOMER_TRX_ALL.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 receipt 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 receipt method name in your batch source, AutoInvoice defaults a value in this column.

AutoInvoice always defaults the receipt 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 receipt 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_ALL.RECEIPT_METHOD_ID

RECEIPT_METHOD_NAME

Enter the name of the receipt 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_ALL.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_ALL.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_ALL.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_ALL.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_ATTRIBUTE1-15. You must enter a value in this column if you entered values in RA_INTERFACE_LINES_ALL.ATTRIBUTE1-15.

Validation: Must exist in
RA_CUSTOMER_TRX_LINES_ALL.INTERFACE_LINE_CONTEXT or

RA_INTERFACE_LINES_ALL.INTERFACE_LINE_CONT
EXT

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_ALL.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_ALL.CUSTOMER_TRX_LINE_ID of the commitment line you are referencing.

Note: Note: An invoice can be attached to only one commitment. Upon import, if an invoice has multiple lines where different commitment line values are provided in the REFERENCE_LINE_ID column, then Receivables creates one or more invoices, accordingly.

Tip: If an invoice has multiple lines but a commitment's balance covers only a partial invoice amount, then Receivables can still create a single invoice upon import. To accomplish this, all lines must have the same commitment line value but, using the PROMISED_COMMITMENT_AMOUNT column, some invoice lines will deplete the commitment's remaining balance while other invoice lines will have an allocated commitment value of zero.

If LINE_TYPE= 'TAX' and this transaction is a credit memo, you must enter the RA_CUSTOMER_TRX_LINES_ALL.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_ALL.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_ALL.CUSTOMER_TRX_LINE_ID.

Destination: RA_CUSTOMER_TRX_LINES_ALL.PREVIOUS_CUSTOM

ER_TRX_LINE_ID if this transaction is a credit memo.
Otherwise,
RA_CUSTOMER_TRX_LINES_ALL.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_ALL.NAME and RA_INTERFACE_LINES_ALL.RELATED_TRX_NUMBER = RA_CUSTOMER_TRX_ALL.TRX_NUMBER and RA_BATCH_SOURCES_ALL.BATCH_SOURCE_ID = RA_CUSTOMER_TRX_ALL.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_ALL.CUSTOMER_TRX_ID

Destination: RA_CUSTOMER_TRX_ALL.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_ALL.NAME and
RA_INTERFACE_LINES_ALL.RELATED_TRX_NUMBER = RA_CUSTOMER_TRX_ALL.TRX_NUMBER and
RA_BATCH_SOURCES_ALL.BATCH_SOURCE_ID = RA_CUSTOMER_TRX_ALL.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_ALL, RA_CUSTOMER_TRX_LINES_ALL, RA_CUST_TRX_LINE_GL_DIST_ALL, AR_PAYMENT_SCHEDULES_ALL, AR_RECEIVABLE_APPLICATIONS_ALL, AR_ADJUSTMENTS_ALL and RA_CUST_TRX_LINE_SALESREPS_ALL.

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.

For more information about rule start dates, see: Determining Dates, *Oracle Receivables*

User Guide.

Validation: None

Destination: RA_CUSTOMER_TRX_LINES_ALL.RULE_START_DATE

RULE_END_DATE

Enter the date that you want to end the accounting rule for this transaction.

This column is required if the accounting rule is either *Daily Revenue Rate, All Periods* or *Daily Revenue Rate, Partial Periods*.

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.

For more information about rule start dates, see: Determining Dates, *Oracle Receivables User Guide*.

Validation: None

Destination: RA_CUSTOMER_TRX_LINES_ALL.RULE_END_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_ALL.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_ALL.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_ALL.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_ALL.SALES_ORDER_REV

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_ALL.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_ALL.SALES_TAX_ID

SET_OF_BOOKS_ID

Optionally enter the ledger ID for this transaction. If no value exists, then Receivables defaults the ledger from the System Options window for the organization that is specified in the ORG_ID column.

Validation: Must exist in
AR_SYSTEM_PARAMETERS_ALL.SET_OF_BOOKS_ID

Destination: RA_CUSTOMER_TRX_ALL.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 earliest shipment date from the transaction you are crediting.

Validation: None

Destination: RA_CUSTOMER_TRX_ALL.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.FREIGHT_CODE =
RA_INTERFACE_LINES_ALL.SHIP_VIA and
ORG_FREIGHT.ORGANIZATION_ID =
RA_INTERFACE_LINES_ALL.WAREHOUSE_ID.
RA_INTERFACE_LINES_ALL.SHIP_VIA must be less than
or equal to 25 characters in length.

Destination: RA_CUSTOMER_TRX_ALL.SHIP_VIA

SOURCE_DATA_KEY1-5

Enter line group attributes that link one or more transaction lines into groups.

Receivables uses groups during line-level cash application.

See: Applying Receipts in Detail, *Oracle Receivables User Guide*.

Validation: None

Destination: RA_CUSTOMER_TRX_LINES_ALL.SOURCE_DATA_KEY
1-5

TAX_CODE

Enter the tax code for this tax line.

If LINE_TYPE = 'CHARGES', or 'FREIGHT', do not enter a value in this column.

If LINE_TYPE = 'LINE', this column is optional.

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 Receivables.

For all other line types and 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_ALL.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_ALL.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 the code 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_ALL.TAX_EXEMPT_REASON_CODE

TAX_EXEMPT_REASON_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 and 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

Important: This column is obsolete and should not be populated.

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_ALL.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_ALL.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_ALL.TERRITORY_ID

TERRITORY_SEGMENT1-20

Assign a Territory Flexfield value for each segment you enable in Receivables. 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

TRANSLATED_DESCRIPTION

The translated description of this transaction line (used for multi-lingual support)

Validation None

Destination RA_CUSTOMER_TRX_LINES_ALL.TRANSLATED_DESCRIPTION

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 in the Run AutoInvoice window, 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_ALL.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_ALL.TRX_NUMBER and
RA_CUSTOMER_TRX_ALL.BATCH_SOURCE_ID

Destination: RA_CUSTOMER_TRX_ALL.TRX_NUMBER and
AR_PAYMENT_SCHEDULES_ALL.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_ALL.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 amount/quantity 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 the value 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_ALL.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_ALL.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: None

Destination: RA_CUSTOMER_TRX_LINES_ALL.DEFAULT_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_ALL.VAT_TAX_ID

WAREHOUSE_ID

This column identifies the ship-from location and can be used to control taxation. Within the US, the Warehouse ID is important when calculating tax on the Origin/Modified Origin state sales tax (outside the US, you can use Tax Groups and Conditions to build a schedule of multiple conditional taxes based on both the ship-from and ship-to County/County/State or Provinces).

Validation None

Destination RA_CUSTOMER_TRX_LINES_ALL.WAREHOUSE_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_ALL.WAYBILL_NUMBER

Table Name: RA_INTERFACE_SALESCREDITS_ALL

This table stores sales 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 field of the System Options window and Allow Sales Credits field in the Transaction Sources window will determine whether you must enter sales credit information. See: AutoAccounting, *Oracle Receivables Implementation Guide* and Defining Receivables System Options, *Oracle Receivables Implementation Guide*.

If you are importing invoices, debit memos and on-account credits and your system option requires a salesperson, you must provide sales credit information, regardless of the value entered in the Allow Sales Credit field for your transaction batch source.

If you are importing credit memos and your system option requires that you enter a salesperson, you can provide sales credit information. If you do not provide sales credit information, AutoInvoice uses sales credit information from the invoice you are crediting. If the invoice you are crediting does not have sales credit information, AutoInvoice creates a 100% 'No Sales Credit' line for this invoice. This sales credit line is then used to determine the sales credit amount for the credit memo.

Regardless of the type of transaction you are importing, if your system option does not require salesperson, but your transaction batch source allows sales credits, you can provide sales credit information. AutoInvoice will validate it and pass this information with your transaction. If your system option does not require salesperson and your transaction batch source does not allow sales credits, do not provide sales 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_ALL.ATTRIBUTE1-1

5

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_ALL.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 Receivables 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 in this column.

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_ALL.CUSTOMER_TRX_LINE_ID

INTERFACE_SALESCREDIT_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_SALESREPS_S.

Validation: None

Destination: RA_CUST_TRX_LINE_SALESREPS_ALL.CUST_TRX_LINE_SALESREP_ID

INTERFACE_STATUS

This column is used by AutoInvoice and should be left null.

Validation: None

Destination: None

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_SALESCREDITS_ALL table for processing.

Validation: None

Destination: None

ORG_ID

Enter the ID of the organization that this transaction belongs to. This column is mandatory in a multiple organization environment.

Validation: AutoInvoice imports transactions whose ORG_ID matches the value of the MO: Operating Unit profile option.

Destination: None.

REQUEST_ID

This column is used by AutoInvoice and should be left null.

Validation: None

Destination: None

SALES_CREDIT_AMOUNT_SPLIT

Enter the sales credit amount for this salesperson. 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 credit percent in your batch source, AutoInvoice defaults a value in this column.

Validation: If the sales credit for this sales credit assignment is of type Quota, the sum of sales credit amounts for a transaction must equal the amount of the transaction.

Destination: RA_CUST_TRX_LINE_SALESREPS_ALL.REVENUE_AMOUNT_SPLIT if the sales credit type is Quota.
RA_CUST_TRX_LINE_SALESREPS_ALL.NON_REVENUE_AMOUNT_SPLIT if the sales credit type is not Quota.

SALES_CREDIT_PERCENT_SPLIT

Enter the sales credit percent for this salesperson. 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 credit type is Quota, the sales credit percentage for a transaction must sum to 100.

SALES_CREDIT_TYPE_ID

Enter the ID of the sales credit type for this sales 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 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 credit type for this sales 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

SALES_GROUP_ID

Enter the sales group ID for this sales credit assignment. This column is optional.

Validation: Must exist in JTF_RS_GROUP_USAGES.GROUP_ID and have JTF_RS_GROUP_USAGES.USAGE = 'SALES'

Destination: RA_CUST_TRX_LINE_SALESREPS_ALL.REVENUE_SAL_ESGROUP_ID or
RA_CUST_TRX_LINE_SALESREPS_ALL.NON_REVENU_E_SALESGROUP_ID

SALESREP_ID

Enter the salesperson ID for this sales 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 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_ALL.SALESREP_ID

SALESREP_NUMBER

Enter the salesperson number for this sales 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

Table Name: 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 that you pass.

Distributions in this table are linked to the appropriate transaction lines in the ra_interface_lines 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. AutoInvoice will then correctly transfer all distributions to RA_CUST_TRX_LINE_GL_DIST_ALL.

ACCOUNT_CLASS

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.

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.
Destination:	RA_CUST_TRX_LINE_GL_DIST_ALL.ACOUNT_CLASS

ACCTD_AMOUNT

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. For imported amounts in the functional currency, AutoInvoice will reject the line if you enter a value in the ACCTD_AMOUNT column that does not equal the line amount.

Validation:	None
Destination:	None

AMOUNT

Enter the amount for this accounting distribution.

If this accounting distribution is for a transaction that 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.

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.

If this line has AMOUNT_INCLUDES_TAX set to Yes, the sales credits and line amounts for this column must include tax.

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.
Destination:	RA_CUST_TRX_LINE_GL_DIST_ALL.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_ALL.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_ALL.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_ALL.COLLECTED_TAX_CCID if tax is deferred; otherwise,
RA_CUST_TRX_LINE_GL_DIST_ALL.CODE_COMBINATION_ID

COMMENTS

Enter comments about this accounting distribution. This column is optional.

Validation: None

Destination: RA_CUST_TRX_LINE_GL_DIST_ALL.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_ALL.CUST_TRX_LINE_GL_DIST_ID

INTERFACE_LINE_ATTRIBUTE1-15

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.

Validation:	None
Destination:	None

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.

Validation:	If you pass lines with global context set this column to 'Global Data Elements'
Destination:	RA_CUSTOMER_TRX_LINES_ALL.INTERFACE_LINE_CONTEXT

INTERFACE_LINE_ID

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.

Validation:	None
Destination:	RA_CUST_TRX_LINE_GL_DIST_ALL.CUSTOMER_TRX_LINE_ID

INTERFACE_STATUS

This column is used by AutoInvoice and should be left null.

Validation:	None
Destination:	None

INTERIM_TAX_CCID

This column identifies the tax account used for deferred tax amounts.

Validation:	None
Destination:	RA_CUST_TRX_LINE_GL_DIST_ALL.CODE_COMBINATION_ID

INTERIM_TAX_SEGMENT1-30

Enter an Accounting Flexfield value for each segment you enable in Receivables. This flexfield represents the Interim (deferred) tax account. 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, the 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
---------------------	------

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_DISTRIBUTIONS_ALL table for processing.

Validation:	None
--------------------	------

Destination:	None
---------------------	------

ORG_ID

Enter the ID of the organization that this transaction belongs to. This column is mandatory in a multiple organization environment.

Validation:	AutoInvoice imports transactions whose ORG_ID matches the value of the MO: Operating Unit profile option.
--------------------	---

Destination:	None.
---------------------	-------

PERCENT

Enter the percent for this accounting distribution.

If this accounting distribution is for a transaction that 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_ALL.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 Receivables. 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, the 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

Table Name: AR_INTERFACE_CONTS_ALL

This table stores information about contingencies that impact revenue recognition for your imported transactions.

See: Event-Based Revenue Management, *Oracle Receivables User Guide*.

CONTINGENCY_ID

Identifies the revenue contingency, according to this table:

Contingency Name	Contingency ID
Explicit Acceptance	2

Contingency Name	Contingency ID
Customer Creditworthiness	3
Doubtful Collectibility	4
Extended Payment Term	5
Cancellation	7
Fiscal Funding Clause	8
Refund	9
Forfeitures	10
Leasing Doubtful Collectibility	12
Impaired Loans	13

Validation: None.

Destination: AR_LINE_CONTS_ALL.CONTINGENCY_ID

EXPIRATION_DATE

Indicates expiration date of contingency. For time-based contingencies, enter either expiration date or expiration days.

Validation: None.

Destination: AR_LINE_CONTS_ALL.CONTINGENCY_CODE

EXPIRATION_DAYS

Indicates expiration period of contingency. For time-based contingencies, enter either expiration date or expiration days.

Validation: None.

Destination: AR_LINE_CONTS_ALL.CONTINGENCY_CODE

EXPIRATION_EVENT_DATE

Indicates the expiration of the contingency removal event.

Validation:	None.
Destination:	AR_LINE_CONTS_ALL.EXPIRATION_EVENT_DATE

INTERFACE_CONTINGENCY_ID

Contingency identifier.

Validation:	None
Destination:	None

ORG_ID

Enter the ID of the organization that this transaction belongs to. This column is mandatory in a multiple organization environment.

Validation:	AutoInvoice imports transactions whose ORG_ID matches the value of the MO: Operating Unit profile option.
Destination:	None.

PARENT_LINE_ID

Identifies the original parent order line from Oracle Order Management. Child invoice lines inherit contingencies from the parent line, and cannot be updated.

Validation:	None
Destination:	None

Table Name: RA_INTERFACE_ERRORS_ALL

This table stores information about interface lines that failed validation and were not imported into Receivables tables. Receivables uses the information in this table to generate the AutoInvoice Validation Report, *Oracle Receivables User Guide*. AutoInvoice identifies all errors for each transaction line, thus reducing multiple validation and correction cycles. When you resubmit AutoInvoice, the program deletes the errors for each line selected for processing. When all of the records have been successfully processed, AutoInvoice purges any remaining data in this table.

Use the Interface Exceptions window to view all of the errors in RA_INTERFACE_ERRORS_ALL. For more information, see: Correcting AutoInvoice Exceptions, *Oracle Receivables User Guide*.

INTERFACE_LINE_ID

If both INTERFACE_SALESCREDIT_ID and INTERFACE_DISTRIBUTION_ID are null,

then the row in RA_INTERFACE_LINES_ALL associated with this INTERFACE_LINE_ID failed validation.

Validation: None

Destination: None

INTERFACE_SALESCREDIT_ID

If this column is not null, then the row in RA_INTERFACE_SALESCREDITS_ALL associated with this INTERFACE_SALESCREDIT_ID failed validation.

Validation: None

Destination: None

INTERFACE_DISTRIBUTION_ID

If this column is not null, then the row in RA_INTERFACE_DISTRIBUTIONS_ALL associated with this INTERFACE_DISTRIBUTION_ID failed validation.

Validation: None

Destination: None

INVALID_VALUE

The invalid value that failed validation displays in this column, if applicable.

Validation: None

Destination: None

LINK_TO_LINE_ID

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. In this case, the column stores the INTERFACE_LINE_ID of the invoice line.

Validation: None

Destination: None

MESSAGE_TEXT

The message text is stored in this column.

Validation: None

Destination: None

Related Topics

Importing Transactions Using AutoInvoice, *Oracle Receivables User Guide*

Using AutoInvoice, *Oracle Receivables User Guide*

Lockbox Table and Column Descriptions

When you submit the Import, Validation, and Post Batch steps of AutoLockbox, Receivables stores receipt information in temporary application tables until it is approved for the next step. For example, the Validation step checks data in the AutoLockbox tables for compatibility with Receivables before passing the information into the Receipt and QuickCash tables. The following sections describe these tables.

Related Topics

Receipt and QuickCash Tables, page B-62

Lockbox Interface Table and Column Descriptions, page B-62

Running AutoLockbox, *Oracle Receivables User Guide*

Receipt and QuickCash Tables

When you run the Validation step, Lockbox transfers receipt data into the following QuickCash tables:

AR_INTERIM_CASH_RECEIPTS_ALL

AR_INTERIM_CASH_RCPT_LINES_ALL

When you run Post QuickCash, the receipt data is transferred from the QuickCash tables to the following Receipt tables:

AR_CASH_RECEIPTS_ALL

AR_RECEIVABLES_APPLICATIONS_ALL

AR_CASH_RECEIPT_HISTORY_ALL

Related Topics

Lockbox Interface Table and Column Descriptions, page B-62

Lockbox Interface Table and Column Descriptions

When you run the Import step of AutoLockbox, Receivables stores receipt data from your bank file in the AR_PAYMENTS_INTERFACE_ALL Lockbox Interface table.

Following is a detailed description of this table.

Each column in the AR_PAYMENTS_INTERFACE_ALL table has important, detailed information you need to successfully run AutoLockbox. The Destination column gives you the interim QuickCash tables and the actual Receivables applications tables to which the data is transferred from the AR_PAYMENTS_INTERFACE_ALL table.

Understanding the AR_PAYMENTS_INTERFACE_ALL Table

This section lists the columns in the AR_PAYMENTS_INTERFACE_ALL table, providing each column's type, source, and destination.

TRANSMISSION_RECORD_ID (NUMBER(15))

- Source - AR_PAYMENTS_INTERFACE_S.NEXTVAL
- Destination - None

CREATION_DATE (DATE)

- 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 - CURRENT SYSTEM DATE
- None

RECORD_TYPE (NOT NULL) (VARCHAR2(2))

- Source - AR_TRANS_RECORD_FORMATS.RECORD_IDENTIFIER
- 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

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 - DERIVED 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)

- 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 TRANSFERRED SUCCESSFULLY
- 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

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))

- 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_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_ACCOUNT_ID (NUMBER(15))

- Source - AP_BANK_ACCOUNT_USES.EXTERNAL_BANK_ACCOUNT_ID

- Destination -

AR_INTERIM_CASH_RECEIPTS.CUSTOMER_BANK_ACCOUNT_ID

ANTICIPATED_CLEARING_DATE (DATE)

- Source - LOCKBOX DATA FILE OR ENTERED BY USER VIA 'MAINTAIN LOCKBOX TRANSMISSION DATA'
- Destination - AR_INTERIM_CASH_RECEIPTS.ANTICIPATED_CLEARING_DATE

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

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

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 '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

INVOICE1-8 (VARCHAR2(50))

- Source - LOCKBOX DATA FILE OR ENTERED BY USER VIA 'MAINTAIN LOCKBOX TRANSMISSION DATA'
- Destination - None

MATCHING1_DATE - MATCHING8_DATE (DATE)

- Source - LOCKBOX DATA FILE OR ENTERED BY USER VIA 'MAINTAIN LOCKBOX TRANSMISSION DATA'
- Destination - None

RESOLVED_MATCHING_NUMBER1-8 (NUMBER)

- Source - PROGRAM DETERMINES IT
- Destination - None

RESOLVED_MATCHING1_DATE - RESOLVED_MATCHING8_DATE (DATE)

- Source - PROGRAM DETERMINES IT
- Destination - None

MATCH_RESOLVED_USING (VARCHAR2(30))

- Source - PROGRAM DETERMINES IT
- Destination - None

RESOLVED_MATCHING1_INSTALLMENT - RESOLVED_MATCHING8_INSTALLMENT (NUMBER)

- Source - PROGRAM DETERMINES IT
- Destination - None

INVOICE1_STATUS - INVOICE8_STATUS (VARCHAR2(30))

- Source - PROGRAM DETERMINES IT
- Destination - None

COMMENTS (NUMBER)

- 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

CUSTOMER_NAME_ALT (VARCHAR2(320))

- Source - LOCKBOX DATA FILE
- Destination - None

CUSTOMER_BANK_NAME (VARCHAR2(320))

- Source - LOCKBOX DATA FILE
- Destination - None

CUSTOMER_BANK_BRANCH_NAME (VARCHAR2(320))

- Source - LOCKBOX DATA FILE
- Destination - None

REMITTANCE_BANK_NAME (VARCHAR2(320))

- Source - PROGRAM DETERMINES IT
- Destination - None

REMITTANCE_BANK_BRANCH_NAME (VARCHAR2(320))

- Source - PROGRAM DETERMINES IT
- Destination - None

BANK_TRX_CODE (VARCHAR2(30))

- Source - PROGRAM DETERMINES IT
- Destination - None

AMOUNT_APPLIED1-8 (NUMBER)

- Source - LOCKBOX DATA FILE OR DERIVED FROM AMOUNT_APPLIED_FROM AND EXCHANGE_RATE
- Destination -
 - AR_INTERIM_CASH_RECEIPTS_ALL.AMOUNT_APPLIED (if a single application)
 - AR_INTERIM_CASH_RCPT_LINES_ALL.PAYMENT_AMOUNT (if multiple applications)

AMOUNT_APPLIED_FROM1-8 (NUMBER)

- Source - LOCKBOX DATA FILE OR DERIVED FROM AMOUNT_APPLIED_FROM AND EXCHANGE_RATE
- Destination -
 - AR_INTERIM_CASH_RECEIPTS_ALL.AMOUNT (if a single application)
 - AR_INTERIM_CASH_RCPT_LINES_ALL.AMOUNT_APPLIED_FROM (if multiple applications)

INVOICE_CURRENCY_CODE1-8 (VARCHAR2(15))

- Source - LOCKBOX DATA FILE OR DERIVED FROM AR_PAYMENT_SCHEDULES_ALL
- Destination -
 - AR_INTERIM_CASH_RECEIPTS_ALL.INVOICE_CURRENCY_CODE (if a single application)
 - AR_INTERIM_CASH_RCPT_LINES_ALL.INVOICE_CURRENCY_CODE (if multiple applications)

TRANS_TO_RECEIPT_RATE1-8 (NUMBER)

- Source - LOCKBOX DATA FILE OR DERIVED FROM AMOUNT_APPLIED_FROM AND EXCHANGE_RATE
- Destination - TRANS_TO_RECEIPT_RATE

CUSTOMER_REFERENCE_1-8 (VARCHAR2(100))

- Source -

- Destination -

CUSTOMER_REASON1-8 (VARCHAR2(30))

- Source -

- Destination -

Assigning Values to Columns

You must assign values to all of the following columns in the AR_PAYMENTS_INTERFACE_ALL table for AutoLockbox to successfully convert data into receipts.

STATUS

Enter the value AR_PLB_NEW_RECORD for all records inserted into this table. The sample SQL*Loader control files Receivables provides 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

Receivables lets you determine what information you want to include in your header, trailer, and receipt records. You can reference any of the above types when you define

the different records for your transmission format. Below are 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.

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. If the transmission format includes the transmission header or trailer, Lockbox counts all records in this transmission. The validated count includes all receipts and detail records transferred to the interim table.
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.

LOCKBOX_NUMBER	Enter the lockbox name or number that your bank specifies. This is the same value that you entered in the
-----------------------	---

	Lockboxes window. 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.

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.
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.

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. Receivables supports AutoLockbox Transmission receipts in different currencies.
EXCHANGE_RATE	Enter the exchange rate you want Receivables 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. If you are using MICR numbers to identify customers, Lockbox requires that this date be equal to or earlier than the date of this AutoLockbox submission; otherwise, the receipts will be unidentified.
RECEIPT_METHOD	Enter the receipt method that you want to associate with this receipt. Receipt methods contain information about your bank, bank account, and receipt accounts. This receipt 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. Receivables uses transit routing number and account number together to identify the customer (MICR number).
ACCOUNT	Enter the bank 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 or 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 with a class of PMT or GUAR. Invoice numbers are optional.
AMOUNT_APPLIED_FROM1-8	If the receipt currency and the transaction currency are different, enter the amount of the receipt to apply in the <i>receipt</i> currency.
INVOICE_CURRENCY_CODE1-8	If the receipt currency and the transaction currency are different, enter the currency of the transaction (optional). If null, AutoLockbox derives this value from AR_PAYMENT_SCHEDULES_ALL. This field is used for cross currency receipt applications.
TRANS_TO_RECEIPT_RATE1-8	If the receipt currency and the transaction currency are different, enter the exchange rate used to convert the receipt to the transaction currency. This value is used for cross currency receipt applications when the receipt and transaction currencies do not have a fixed exchange rate.
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, Receivables 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 can provide invoice numbers without specifying the amount applied to each of these invoices. If you provide invoice numbers without specifying the amount applied to each invoice, Receivables applies the receipt to the invoices starting with the oldest receipt schedule first. The value of the amount_applied column 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. If the receipt currency and the transaction currency are different, enter the amount of the receipt to apply in the <i>transaction</i> currency.
COMMENTS	Enter any free-form comments about this receipt. Receivables stores this data, but does not display these comments in any of the receipt entry windows.
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 windows.
BILL_TO_LOCATION	To associate receipts with specific customer sites, enter the billing address for this receipt and include billing location in your transmission format. If the system option Require Billing Location for Receipts is set to Yes, you must enter a value here. In addition, you can set the Require Billing Location field to Yes in the Lockboxes window to require a billing location for a specific lockbox. The value of this field in the Lockboxes window will override the option at the system level. See: Lockboxes, <i>Oracle Receivables Implementation Guide</i> .
CUSTOMER_BANK_NAME	The name of the customer's bank.
CUSTOMER_BANK_BRANCH_NAME	The name of the customer's bank branch.
REMITTANCE_BANK_NAME	The name of the bank that received the payment.

REMITTANCE_BANK_BRANCH_NAME	The name of the bank branch that received the payment.
------------------------------------	--

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. An overflow record can have up to eight invoice applications.

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	Receivables uses this column to indicate overflow records for the current receipt. You determine your overflow indicator in your 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.
AMOUNT_APPLIED_FROM1-8	If the receipt currency and the transaction currency are different, enter the amount of the receipt to apply in the <i>receipt</i> currency.
INVOICE_CURRENCY_CODE1-8	If the receipt currency and the transaction currency are

different, enter the currency of the transaction (optional). If null, AutoLockbox derives this value from AR_PAYMENT_SCHEDULES_ALL. This field is used for cross currency receipt applications.

TRANS_TO_RECEIPT_RATE 1-8	If the receipt currency and the transaction currency are different, enter the exchange rate used to convert the receipt to the transaction currency. This value is used for cross currency receipt applications when the receipt and transaction currencies do not have a fixed exchange rate.
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 can find a list of valid values in AR_PAYMENT_SCHEDULES.TRX_NUMBER. Do not look at transactions with a class of 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 Receivables 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, Receivables applies the receipt to the invoices starting with the oldest receipt first. The value of the amount applied column must be on the same record as the invoice number to which the receipt amount is applied.

System Assigned Columns

Receivables assigns values to the columns listed in the table below during the import process. Your import file must leave these columns blank.

Column Name	Type
TRANSMISSION_RECORD_ID	NUMBER

Column Name	Type
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

Seeded Match Rules

Seeded Search Match Rules

Two seeded match rules, SAMPLE: BASIC SEARCH RULE and SAMPLE: ADVANCED SEARCH RULE, are provided for you to use for the Customer Search page. These rules let you search with the same criteria as in a non-DQM search, but with the robust Data Quality Management matching functionality.

You set either of these two seeded match rules, or one that you define yourself using the Data Quality Management Search Profile Options. See:

- Setting Up DQM, *Oracle Receivables Implementation Guide*.
- DQM Deployment Category, *Oracle Trading Community Architecture Administration Guide*.

The acquisition portion of the match rule determines the displayed search criteria and potential matches. Each acquisition attribute corresponds to a search criterion. The scoring portion scores and ranks the search results.

Prerequisite

- Run the DQM Compile All Rules Program.

See also: DQM Compile All Rules Program, *Oracle Trading Community Architecture Administration Guide*.

Related Topics

Match Rules Overview, *Oracle Trading Community Architecture Administration Guide*

SAMPLE: BASIC SEARCH RULE

This match rule provides search criteria for performing a basic DQM search.

Acquisition

This table shows the seeded attributes and transformation functions for the acquisition part of the matching process.

Attribute Name	Entity	Filter	Attribute Match	Transformation Name
Name	Party	No	Match All Attributes	SOUNDEX WR + CLEANSE
Registry ID	Party	No	Match All Attributes	EXACT
Address	Address	No	Match All Attributes	WR ADDRESS + CLEANSE
City	Address	No	Match All Attributes	CLEANSE
State	Address	No	Match All Attributes	WR STATE
Country	Address	No	Match All Attributes	EXACT
All Account Names	Party	No	Match All Attributes	CLEANSE
Contact Name	Contact	No	Match All Attributes	SOUNDEX WR PERSON + CLEANSE
Phone Number Flexible Format	Contact Point	No	Match All Attributes	EXACT
e-mail Address	Contact Point	No	Match All Attributes	CLEANSE (EMAIL)
Site Number	Address	No	Match All Attributes	EXACT
All Account Numbers	Party	No	Match All Attributes	EXACT (NUMBER)

Scoring

This table shows the seeded thresholds for the scoring part of the matching process.

Threshold	Value
Match Threshold	450
Override Threshold	
Automatic Merge Threshold	

This table shows the seeded attributes and transformations for the scoring part of the matching process.

Attribute Name	Entity	Score	Transformation Name	Weight (%)	Type	Similarity (%)
Name	Party	50	WR NAMES + CLEANSE	100	Exact	
			SOUNDEX	70	Exact	
Registry ID	Party	100	EXACT	100	Exact	
Address	Address	70	WR ADDRESS	100	Exact	
			WR ADDRESS + CLEANSE	70	Exact	
City	Address	20	CLEANSE	70	Exact	
State	Address	10	WR STATE	100	Exact	
Country	Address	10	EXACT	100	Exact	
All Account Names	Party	60	WR NAMES	100	Exact	
			WR NAMES + CLEANSE	80	Exact	
Contact Name	Contact Point	50	WR PERSON + CLEANSE	100	Exact	
			SOUNDEX	70	Exact	

Attribute Name	Entity	Score	Transformation Name	Weight (%)	Type	Similarity (%)
Phone Number Flexible Format	Contact Point	80	EXACT	100	Exact	
e-mail Address	Contact Point	80	EXACT (EMAIL)	100	Exact	
			CLEANSE (EMAIL)	70	Exact	
Site Number	Address	50	EXACT	100	Exact	
All Account Numbers	Party	90	EXACT (NUMBER)	100	Exact	

SAMPLE: ADVANCED SEARCH RULE

This match rule provides search criteria for performing an advanced DQM search.

Acquisition

This table shows the seeded attributes and transformation functions for the acquisition part of the matching process.

Attribute Name	Entity	Filter	Attribute Match	Transformation Name
Name	Party	No	Match All Attributes	SOUNDEX WR NAMES + CLEANSE
Registry ID	Party	No	Match All Attributes	EXACT
Tax Name	Party	No	Match All Attributes	CLEANSE
Party Type	Party	Yes	Match All Attributes	EXACT
Category Code	Party	No	Match All Attributes	EXACT
SIC Code	Party	No	Match All Attributes	EXACT

Attribute Name	Entity	Filter	Attribute Match	Transformation Name
SIC Code Version	Party	No	Match All Attributes	EXACT
Tax Registration Num	Party	No	Match All Attributes	EXACT
City	Address	No	Match All Attributes	CLEANSE
State	Address	No	Match All Attributes	WR STATE
Postal Code	Address	No	Match All Attributes	EXACT
County	Address	No	Match All Attributes	CLEANSE
Province	Address	No	Match All Attributes	CLEANSE
Country	Address	No	Match All Attributes	EXACT
All Account Names	Party	No	Match All Attributes	WR NAMES + CLEANSE
Phone Number	Contact Point	No	Match All Attributes	EXACT
All Account Numbers	Party	No	Match All Attributes	EXACT (NUMBER)
Reference Use Flag	Party	No	Match All Attributes	EXACT
Corporation Class	Party	No	Match All Attributes	EXACT

Scoring

This table shows the seeded thresholds for the scoring part of the matching process.

Threshold	Value
Match Threshold	480
Override Threshold	

Threshold	Value
Automatic Merge Threshold	

This table shows the seeded attributes and transformation functions for the scoring part of the matching process.

Attribute Name	Entity	Score	Transformation Name	Weight (%)	Type	Similarity (%)
Name	Party	50	WR NAMES + CLEANSE	80	Exact	
Name	Party	50	WR NAMES	100	Exact	
Name	Party	50	SOUNDEX	60	Exact	
Registry ID	Party	100	EXACT	100	Exact	
Tax Name	Party	30	CLEANSE	100	Exact	
Reference Use Flag	Party	20	EXACT	100	Exact	
Category Code	Party	10	EXACT	100	Exact	
SIC Code	Party	30	EXACT	100	Exact	
SIC Code Version	Party	10	EXACT	100	Exact	
Tax Registration Num	Party	80	EXACT	70	Exact	
Corporation Class	Party	30	EXACT	100	Exact	
City	Address	30	CLEANSE	70	Exact	
State	Address	20	WR STATE	80	Exact	

Attribute Name	Entity	Score	Transformation Name	Weight (%)	Type	Similarity (%)
Postal Code	Address	30	EXACT	100	Exact	
County	Address	20	CLEANSE	70	Exact	
Province	Address	10	CLEANSE	70	Exact	
Country	Address	30	EXACT	100	Exact	
All Account Names	Party	50	EXACT	100	Exact	
Phone Area Code	Contact Point	30	EXACT	100	Exact	
All Account Numbers	Party	100	EXACT (NUMBER)	100	Exact	
Phone Number	Contact Point	50	EXACT	100	Exact	
Phone Country Code	Contact Point	10	EXACT	100	Exact	

D

XML Transactions

XML Receivables Documents Mapping

This appendix provides the mapping for the XML messages used in the XML Invoices process.

For more information about sending XML messages, that contain Receivables documents, to customers, see: XML Receivables Documents, *Oracle Receivables User Guide*.

Related Topics

Process Invoice XML Message Map, page D-1

Confirm BOD Message Map, page D-5

Transaction Limitations, page D-6

Process Invoice XML Message Map

The XML invoices process uses the Open Applications Group Process Invoice DTD called 171_process_invoice_002.dtd (version 7.2.1).

The following table shows you the mapping between Receivables database columns and the elements of the Process Invoice XML message for the invoice header.

INVHEADER	Target (XML)	Source
	AMOUNT.DOCUMENT.T.VALUE	AR_XML_INVOICE_V.amount
	AMOUNT.DOCUMENT.T.CURRENCY	AR_XML_INVOICE_V.currency_code

INVHEADER	Target (XML)	Source
	DATETIME.DOCUMENT	AR_XML_INVOICE_V.trx_date
	DOCUMENTID	AR_XML_INVOICE_V.trx_number
	PAYMETHOD	AR_XML_INVOICE_V.payment_method
	PARTNER.NAME	AR_XML_INVOICE_V.supplier_name
	PARTNER.PARTNERID	AR_XML_INVOICE_V.supplier_code
	PARTNER.PARTNERTYPE	Supplier
	PARTNER.PARTNERIDX	AR_XML_INVOICE_V.supplier_code
	PARTNER.CONTACT.NAME	AR_XML_INVOICE_V.salesrep
	PARTNER.CONTACT.EMAIL	AR_XML_INVOICE_V.salesrep_email
	DOCUMNTREFDOCTYPE	AR_XML_INVOICE_V.document_type
	DOCUMNTREFDOCUMENTID	AR_XML_INVOICE_V.reference_number
	DOCUMNTREFPARTNRID	AR_XML_INVOICE_V.supplier_code
	DOCUMNTREFPARTNERTYPE	Supplier
one or more	PYMTTERMAMOUNTDISCNTT	AR_XML_PAYMENT_TERMS_V.discount_amount
one or more	PYMTTERMDATETIMEDISCNT	AR_XML_PAYMENT_TERM_V.discount_date

INVHEADER	Target (XML)	Source
one or more	PYMTTERM.DATETIME.DUE	AR_XML_PAYMENT_TERMS_V.due_date
one or more	PYMTTERM.DATETIME.PYMTTERM	AR_XML_PAYMENT_TERMS_V.terms_start_date_effective
one or more	PYMTTERM.DESCRIPTN	AR_XML_PAYMENT_TERMS_V.terms_description
one or more	PYMTERRM.TERMID	AR_XML_PAYMENT_TERMS_V.terms_name
one or more	PYMTERRM.USERAREA.AMOUNT	AR_XML_PAYMENT_TERMS_V.due_amount

The following table shows you the mapping between Receivables database columns and the elements of the Process Invoice XML message for the invoice charges.

INVCHARGE	Target (XML)	Source
zero or more	AMOUNT.EXTENDED.T	AR_XML_INVOICE_CHARGE_V.charge_amount
zero or more	CHARGETYPE	AR_XML_INVOICE_CHARGE_V.line_type
zero or more	DESCRIPTION	AR_XML_INVOICE_CHARGE_V.description
zero or more	LINENUM	AR_XML_INVOICE_CHARGE_V.line_number

The following table shows you the mapping between Receivables database columns and the elements of the Process Invoice XML message for the invoice lines.

INVLIN	Target (XML)	Source
	AMOUNT.EXTENDED.T	AR_XML_INVOICE_LINE_V.line_amount

INVLIN	Target (XML)	Source
	OPERAMT.UNIT.T	AR_XML_INVOICE_LINE_V.unit_selling_price
	QUANTITY.ITEM	AR_XML_INVOICE_LINE_V.quantity
	LINENUM	AR_XML_INVOICE_LINE_V.line_number
	DESCRIPTN	AR_XML_INVOICE_LINE_V.description
	ITEM	AR_XML_INVOICE_LINE_V.item.description
	ITEMX	AR_XML_INVOICE_LINE_V.item
	UNIT	AR_XML_INVOICE_LINE_V.uom
	DOCUMNTREFDOCTYPE	AR_XML_INVOICE_LINE_V.line_type
	DOCUMNTREFDOCUMENTID	AR_XML_INVOICE_LINE_V.reference_number
	DOCUMNTREFPARTNRID	AR_XML_INVOICE_LINE_V.supplier_code
	DOCUMNTREFPARTNRTYPE	Supplier
	DOCUMNTREFDOCTYPE	PurchaseOrder
	DOCUMNTREFDOCUMENTID	AR_XML_INVOICE_LINE_V.po_number
zero or more	INVCHARGE.AMOUNT.EXTENDED.T	AR_XML_INVOICE_CHARGE_V.charge_amount
zero or more	INVCHARGE.CHARGETYPE	AR_XML_INVOICE_CHARGE_V.line_type
zero or more	INVCHARGE.DESCRIPTION	AR_XML_INVOICE_CHARGE_V.description
zero or more	INVCHARGE.LINENUM	AR_XML_INVOICE_CHARGE_V.line_number
zero or more	INVTA.XAMOUNT.TAX.T	AR_XML_INVOICE_TAX_V.tax_amount

INVLINE	Target (XML)	Source
zero or more	INVTAX.AMOUNT.TAXBASE.T	AR_XML_INVOICE_TAX_V.taxable_amount
zero or more	INVTAX.QUANTITY.PERCENT	AR_XML_INVOICE_TAX_V.tax_rate
zero or more	INVTAX.DESCRIPTN	AR_XML_INVOICE_TAX_V.description
zero or more	INVTAX.LINENUM	AR_XML_INVOICE_TAX_V.line_number
zero or more	INVTAX.TAXCODE	AR_XML_INVOICE_TAX_V.tax_code

Confirm BOD Message Map

The message map for the Confirm_BOD XML message used by the XML Receivables Documents feature is Confirm_BOD (XML - XML, Inbound) and the DTD is 002_confirm_bod_004.dtd.

The Confirm BOD has two control areas. One has the information for the Confirm BOD. The second is an exact copy of the control area from the Process Payment XML message. The second control area provides the context of the Confirm BOD.

The following table shows the mapping of the Confirm BOD XML message.

CONFIRM	Source (XML)	Value	Target (PLSQL)
	STATUSLVL	00: Success 10: AR Invoice Failure	AR_CONFIRMATION.initiate_confirmation_process.P_STATUS
	DESCRIPTIN		
	ORIGREF		AR_CONFIRMATION.initiate_confirmation_process.P_ID. ar_document_transfers.document_transfer_id:ra_customer_trx.customer_trx_id:ra_customer_tx_lines.customer_trx_line_id
			Passed from Receivables to your customer's payables system.

CONFIRM	Source (XML)	Value	Target (PLSQL)
zero or more	CONFIRMMMSG.REASONCODE		AR_CONFIRMATION.initiate_confirmation_process.P_REASON_CODE
zero or more	CONFIRMMMSG.DESCRIPTION	Invoice InvoiceLine CreditMemo CreditMemo Line DebitMemo DebitMemo Line	AR_CONFIRMATION.initiate_confirmation_process.P_DESCRIPTION

Transaction Limitations

This feature has certain limitations. They include the following:

- You can only send invoices, debit memos, credit memos, chargebacks, and deposits as XML documents.
- To *disable* the delivery of XML invoice documents to a customer, you must remove the customer's bill-to site from the Trading Partner Setup window in XML Gateway.

Index

A

account derivation rules, A-12
accounting
 account derivation rules, A-12
 accounting event class options, A-4
 application accounting definitions, A-15
 applications, A-2
 event classes and types, A-3
 event entities, A-2
 journal lines definitions, A-13
 journal line types, A-6
 predefined data, A-1
 process categories, A-4
 sources and assignments, A-5
 subledger accounting methods, A-17
accounting event class options, A-4
Adjustment API
 Ar_Adjust_pub.Approve_Adjustment, 2-13
 Ar_Adjust_pub.Create_Adjustment, 2-4
 Ar_Adjust_pub.Create_Linelevel_Adjustment, 2-25
 Ar_Adjust_pub.Modify_Adjustment, 2-18
 Ar_Adjust_pub.Reverse_Adjustment, 2-23
 messages, 2-28
 overview, 2-1
 usage, 2-1
APIs
 Adjustment API, 2-1
 Adjustment API usage, 2-1
 Credit Memo Application API, 4-1
 Credit Memo Application API usage, 4-1

Credit Memo Approval and Creation API, 3-1
Credit Memo Approval and Creation API usage, 3-2
Deposit API, 5-1
Deposit API usage, 5-1
features, 1-1
Invoice Creation API, 6-1
Invoice Creation API usage, 6-2
Prepayments API, 7-1
Prepayments API usage, 7-2
Receipt API, 8-1
Receipt API usage, 8-2
Revenue Adjustment API, 9-1
Revenue Adjustment API usage, 9-2
solution outline, 1-2
application accounting definitions, A-15
applications, A-2
AutoInvoice
 AR_INTERFACE_CONTS_ALL, B-58
 RA_INTERFACE_DISTRIBUTIONS_ALL, B-53
 RA_INTERFACE_ERRORS_ALL, B-60
 RA_INTERFACE_LINES_ALL, B-1
 RA_INTERFACE_SALESCREDITS_ALL, B-49
 table and column descriptions, B-1
AutoLockbox
 AR_PAYMENTS_INTERFACE_ALL, B-63
 assigning values to overflow records, B-79
 interface tables and column descriptions, B-62
 tables and column descriptions, B-62

C

columns
AutoInvoice columns and tables, B-1
AutoLockbox columns and tables, B-62

Credit Memo Application API
ar_cm_application_pub.activity_application, 4-1
ar_cm_application_pub.activity_unapplication, 4-11
messages, 4-15
overview, 4-1
usage, 4-1

Credit Memo Approval and Creation API
AR_CM_API_PUB.Apply_On_Account, 3-13
AR_CM_API_PUB.Unapply_On_Account, 3-13
AR_CREDIT_MEMO_API_PUB.Create_Request, 3-2
AR_CREDIT_MEMO_API_PUB.Get_Request_Status, 3-9
messages, 3-13
overview, 3-1
usage, 3-2

customers
seeded match rules for search, C-1

D

Data Quality Management
advanced search rule, C-4
basic search rule, C-1
seeded match rules, C-1

Deposit API
AR_DEPOSIT_API_PUB.Create_deposit, 5-2
AR_DEPOSIT_API_PUB.insert_non_rev_sales_credit, 5-27
messages, 5-32
overview, 5-1
usage, 5-1

E

event classes and types, A-3
event entities, A-2

I

Invoice Creation API
AR_INVOICE_API_PUB, 6-2

overview, 6-1
usage, 6-2

J

journal lines definitions, A-13
journal line types, A-6

M

match rules
advanced search rule, C-4
basic search rule, C-1
seeded, C-1

messages
XML messages mapping, D-1

Messages
Adjustment API, 2-28
Credit Memo Application API, 4-15
Credit Memo Approval and Creation API, 3-13
Deposit API, 5-32
Prepayments API, 7-11
Receipt API, 8-137
Revenue Adjustment API, 9-31

O

Oracle Subledger Accounting
account derivation rules, A-12
accounting event class options, A-4
application accounting definitions, A-15
applications, A-2
event classes and types, A-3
event entities, A-2
journal lines definitions, A-13
journal line types, A-6
predefined data, A-1
process categories, A-4
sources and assignments, A-5
subledger accounting methods, A-17

P

parties
seeded match rules for search, C-1

Prepayments API
AR_PREPAYMENTS_PUB.Create_Prepayment, 7-2

AR_PREPAYMENTS_PUB.Get_Installment, 7-9
messages, 7-11
overview, 7-1
usage, 7-2
process categories, A-4

Q

QuickCash
tables, B-62

R

Receipt API
Ar_receipt_api_pub.activity_application, 8-75
Ar_receipt_api_pub.activity_unapplication, 8-83
Ar_receipt_api_pub.Apply, 8-20
Ar_receipt_api_pub.Apply_on_account, 8-60
Ar_receipt_api_pub.apply_open_receipt, 8-112
Ar_receipt_api_pub.apply_other_account, 8-101
Ar_receipt_api_pub.Create_and_apply, 8-34
Ar_receipt_api_pub.Create_apply_on_acc, 8-121
Ar_receipt_api_pub.Create_cash, 8-3
Ar_receipt_api_pub.Create_misc, 8-87
Ar_receipt_api_pub.Reverse, 8-69
Ar_receipt_api_pub.Unapply, 8-54
Ar_receipt_api_pub.Unapply_on_account, 8-65
Ar_receipt_api_pub.unapply_open_receipt, 8-119
Ar_receipt_api_pub.unapply_other_account, 8-108
messages, 8-137
overview, 8-1
usage, 8-2

receipts
QuickCash
tables, B-62

Revenue Adjustment API

AR_RevenueAdjust_PUB.Add_Non_Revenue_Sales_Credits, 9-23
AR_RevenueAdjust_PUB.Earn_Revenue, 9-16
AR_Revenueadjust_PUB.Record_Customer_Acceptance, 9-27

AR_RevenueAdjust_PUB.Transfer_Sales_Credits, 9-18
AR_RevenueAdjust_PUB.Unearn_Revenue, 9-2
AR_Revenueadjust_PUB.Update_Contingency_Expirations, 9-28
messages, 9-31
overview, 9-1
usage, 9-2

S

seeded match rules, C-1
sources and assignments, A-5
subledger accounting methods, A-17

T

tables
AR_INTERFACE_CONTS_ALL, B-58
AutoInvoice tables and column descriptions, B-1
AutoLockbox tables and columns, B-62
Lockbox interface table and columns, B-62
RA_INTERFACE_DISTRIBUTIONS_ALL, B-53
RA_INTERFACE_ERRORS_ALL, B-60
RA_INTERFACE_LINES_ALL, B-1
RA_INTERFACE_SALESCREDITS_ALL, B-49
receipt and QuickCash tables, B-62

X

XML
limitations, D-6
mapping, D-1

