Oracle Financial Services Revenue Management and Billing
OR
Oracle Insurance Revenue Management and Billing

Version 7.0.0.0.0

Business Process Guide

Revision 13.5

G16855-06

November 2024

ORACLE

Notices

Oracle Financial Services Revenue Management and Billing/Oracle Insurance Revenue Management and Billing Version 7.0.0.0.0 Business Process Guide

Note: To improve the content readability, the above two products are collectively referred to as Oracle Revenue Management and Billing throughout this document.

G16855-06

Copyright Notice

Copyright © 2024, 2025, Oracle and/or its affiliates.

License Restrictions

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 de-compilation of this software, unless required by law for interoperability, is prohibited.

Warranty Disclaimer

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.

Restricted Rights Notice

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

> U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are

defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

Hazardous Applications Notice

This software or hardware 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 that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Trademark Notice

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

Third-Party Content, Products, and Services Disclaimer

This software or hardware and documentation may provide access to or information about 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 unless otherwise set forth in an applicable agreement between you and Oracle. 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, except as set forth in an applicable agreement between you and Oracle.

Preface

About this Document

This document explains how to maintain the demographic, geographic, and financial objects (i.e. accounts) of a customer. It explains how to manage a customer's bills, payments, adjustments, credits, collections processing, statements and deposits in Oracle Revenue Management and Billing (ORMB). It also describes the financial transactions, case management, sales and marketing functions, rates engine, quotations, loans, how to monitor and execute job streams, and how to manage workflows, notifications and overdue processing. In addition, it explains how to extract the data from the system using an extract template.

The features listed and described in this document can be used in both financial services and health insurance domains.

Intended Audience

This document is intended for the following audience:

- End-users
- System Administrators
- Consulting Team
- Implementation Team
- Development Team

Organization of the Document

The information in this document is organized into the following chapters:

Chapter Number	Chapter Name	Chapter Description
Chapter 1	Person Information	Explains how to maintain the persons, account, contracts, effective dated addresses, and customer contacts in the system. It also describes the different ways in which you can print letters in the system.
Chapter 2	Billing	Describes the billing process and explains how to maintain the billable charges, bills, and bill segments. It also explains how to maintain the charge types and invoice frequencies in the system.
Chapter 3	Payments	Describes the payments, automatic payments, and automatic refunds. It explains how to maintain the tender controls, deposit controls, payment events, and payments in the system. In addition, it explains how to interface payments from the external sources.
Chapter 4	Adjustments	Describes adjustments and how and when to use the adjustments. It also explains how to maintain the adjustments in the system. In addition, it explains how to interface adjustments from the external sources.
Chapter 5	Credit & Collections	Describes the collection and write off processes. It explains how to maintain the collection and write off processes in the system. In addition, it explains how to create the payment arrangements and promise to pay in the system.

Chapter Number	Chapter Name	Chapter Description
Chapter 6	Financial Transactions	Describes the different types of financial transactions and how to view the financial transactions generated in the system. It describes match events and how to view the match events in the system. In addition, it explains how to view the account financial history, account bill/payment history, contract financial history, and billed revenue in the system.
Chapter 7	Statements	Describes how to generate statements for a customer. It also describes statement constructs and explains how to maintain statement constructs and statements in the system.
Chapter 8	Sales & Marketing	Describes campaigns, packages, and orders. It also explains how to maintain the campaigns, packages, and orders in the system.
Chapter 9	Rates	Explains how to maintain the rate schedules, rate versions, and rate components in the system. It also explains how to maintain the service quantity identifiers and bill factors in the system.
Chapter 10	Quotes	Describes quotation. It also explains how to maintain and print the quotes in the system.
Chapter 11	Service Credits	Describes service credit membership and service credit event. It also explains how to maintain the service credit memberships and service credit events in the system.
Chapter 12	Loans	Describes how to work with loans in the system. It also explains how to maintain the loans in the system.
Chapter 13	Deposits	Describes how to maintain the different types of deposits in the system.
Chapter 14	Case Management	Explains how to maintain the cases in the system.
Chapter 15	Job Streams	Describes how to execute and monitor the job streams.
Chapter 16	Workflow and Notifications	Describes how to upload and download notifications in the system. It also explains how to maintain the workflow process in the system.
Chapter 17	Overdue Financial Obligations	Explains how to maintain the overdue process and payment arrangement for bills in the system.
Chapter 18	Data Extractor	Describes how to extract data from the system in the CSV format and feed the extracted data from the system to other external systems like data warehousing, reporting systems, analytics etc.
Chapter 19	File Upload Interface	Describes how to upload files in various formats using the **File Upload Interface** feature.
Chapter 20	Extract Template	Explains how to extract the data from the system in various formats using an extract template.
Chapter 21	Batch Scheduler	Explains how to create, schedule, and monitor a job for one or more batch processes in the system.
Appendix A	Administration	Describes how to set various feature configurations in the system.

Conventions

The following conventions are used across this document:

Copyright © 2024, Oracle and/or its affiliates.

Convention	Meaning
boldface	Boldface indicates graphical user interface elements associated with an action, or terms defined in the text.
italic	Italic indicates a document or book title.
`Monospace`	Monospace indicates information that an end-user needs to enter in the application.

Related Documents

You can refer to the following documents for more information:

Document Name	Description
Oracle Revenue Management and Billing Version 7.0.0.0.0 Release Notes	Provides a brief description about the new features, enhancements, and UI level changes introduced in the current release. The new features and enhancements are classified under three distinct heads - Generic (which is intended for both domains), Financial Services, and Health Insurance. It also contains high-level information about the supported platforms, framework upgrade, supported upgrades, and documentation updates. In addition, it contains a list of discontinued features, planned deprecation notices, customer bugs fixed or forward ported and known issues in the current release.
Oracle Utilities Application Framework Business Process Guide	Explains how to get acquainted with the user interface. It explains the different types of pages or portals that you may come across in the application. It explains how to set the user preferences and how to create, manage, assign, and complete a To Do in the application. It also explains how to submit reports and view historic reports in the application.
Oracle Utilities Application Framework Administrative Guide	Explains the general, security, user, designing, developing, and scripting options available in Oracle Utilities Application Framework (OUAF). It describes the user interface, database, configuration, and reporting tools available in OUAF. In addition, it provides information about the application viewer, how to configure incoming and outgoing messages, and how to integrate Lightweight Directory Access Protocol (LDAP), Oracle Identity Manager (OIM), and Batch Scheduler with Oracle Revenue Management and Billing (ORMB).
Oracle Revenue Management and Billing Administrative Guide	Explains how to configure various features and functionalities in Oracle Revenue Management and Billing (ORMB). For example, billing, payments, adjustments, financial transactions, credits, collections processing, loans, service credits, background processes, quotations, case management, security, overdue processing, batch scheduler, workflow and notifications, etc. The information available in this document can be used in both financial services and health insurance domains.
Oracle Revenue Management and Billing Banking User Guide	Describes various features which are available for the financial services business. For example, customer registration, Customer 360° view, invoicing group, pricing management, multi-currency accounts, currency conversion, construct based billing and settlement, trial billing, product lifecycle management, subscription billing, mass pricing update, accrual, foreign exchange gain

Copyright © 2024, Oracle and/or its affiliates.

Document Name	Description
	loss, transaction feed management, upload validated payment and adjustment data, freeze payments on notification, payment request, offset request, funding request, hold request, refund/ write off request, dispute request, upload request, earnings credit rate, payment agreement request, invoice request, deal pricing management, etc. It describes all screens related to these features and explains how to perform various tasks related to the feature in the application.
Oracle Revenue Management and Billing Insurance User Guide	Describes various features which are available for the following three lines of health insurance business - fully-insured group, self-funded, and individual. For example, customer registration, Customer 360° view, pricing management, trial billing, transaction feed management, upload validated payment and adjustment data, freeze payments on notification, payment request, offset request, funding request, hold request, refund/write off request, upload request, payment agreement request, invoice request, inbound message processing, reconciliation, entity audit, premium repricing, etc. It describes all screens related to these features and explains how to perform various tasks related to the feature in the application.
Oracle Revenue Management and Billing TFM - Batch Execution Guide	Explains the sequence in which the batches should be executed while performing various tasks in the Transaction Feed Management (TFM) module. It provides detailed information about each TFM batch and its parameters. It also indicates the restart and multi-threading ability of each batch. In addition, it recommends values for various parameters which can be used for tuning batch performance as per the available hardware.
Oracle Revenue Management and Billing Batch Guide	Provides detailed information about various batches which are used in different modules, such as billing, payments, financial transaction, pricing management, funding request, offset request, hold request, upload request, inbound message, payment agreement request, accruals, earnings credit rate, ILM, deferred revenue recognition, reconciliation, garbling, repricing, entity audit, statements, etc. It also contains information about the batch parameters and the batch restart and multi-threading abilities.

Change Log

The following table lists the updates made to the *Oracle Revenue Management and Billing Business Process User Guide*.

Revision	Last Update	Chapter Name	Updated Section	Comments
13.1	03-02-2025	Data Extractor	-	Added Information
13.2	18-06-2025	Data Extractor	-	Updated Information
		File Upload Interface	File Request Type	Updated Information

Copyright © 2024, Oracle and/or its affiliates.

Revision	Last Update	Chapter Name	Updated Section	Comments
			File Upload Dashboard	Updated Information
			File Record Detail	Updated Information
			File Management System	Updated Information
13.3	25-06-2025	Data Extractor	-	Updated Information
13.4	04-08-2025	Person Information	Billing Preference	Removed Section
		Data Extractor	-	Updated Information
		File Upload Interface	File Record Details	Updated Information
			File Management System	Updated Information
13.5	11-08-2025	File Upload Interface	Setting the C1-FLUPLD Feature Configuration	Updated Information
			Setting the File Upload Interface Master Configuration	Updated Information
			Defining a File Request Type	Updated Information
			Decrypting a File	Added Information

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Copyright © 2024, Oracle and/or its affiliates.

Contents

Notices i

Preface: About this Document iii
- Intended Audience iii
- Organization of the Document iii
- Conventions iv
- Related Documents v
- Change Log vi
- Access to Oracle Support vii

Chapter 1: Person Information 1
- Understanding The V 2
  - Persons 2
  - Accounts 2
  - Contracts 3
  - How To Set Up Customer Hierarchies 3
- Maintaining The V 4
  - Maintaining Persons 4
    - Person - Main 4
    - Person - Contact Information 6
    - Person - Characteristics 8
    - Person - Persons 8
    - Person - Web Self Service 9
    - Person - Person To Person Characteristics 9
    - Person - Usage Amount/Counter 10
    - Person - Deal Information 11
    - Person - Address 12
  - Maintaining Accounts 12
    - Account - Main 12
    - Account - Auto Pay 15
    - Account - Persons 17
    - Account - Financial Balances 20
    - Account - Bill Messages 20
    - Account - C&C 21
    - Account - Budget 22
    - Account - Deposits 23
    - Account - Characteristics 24
    - Account - Alerts 25
    - Account - Account Identifiers 25
    - Account - Usage Amount/Counter 25
  - Account - Contracts For A Debt Class 26
  - Account Closure 27
  - AutoPay Add/Update, Delete, and View Through Inbound Web Service 28
    - Prerequisites 29
    - C1-AddUpdateAutoPay Business Service 29
    - C1-DeleteAutoPay Business Service 45
    - C1-ViewAutoPay Business Service 52

Copyright © 2024, Oracle and/or its affiliates.

Using The Account / Person Replicator....63
Information Replicated for New Persons....63
Information Replicated for New Accounts....64
Maintaining Contracts....64
The Lifecycle Of A Contract....64
Contract - Main Information....66
Contract - Rate Info....69
Contract - Billing Scenarios....71
Contract - Chars, Qty & Rec. Charges....71
Contract - Miscellaneous....72
Automatic Contract Creation....73
Automatic Contract Creation Extendable Lookup....73
Extendable Lookup Search....73
Searching for the C1-CONTRACTTYPES Extendable Lookup....74
Viewing the Values of the C1-CONTRACTTYPES Extendable Lookup....75
Defining a Value for the C1-CONTRACTTYPES Extendable Lookup....75
Extendable Lookup (Automatic Contract Creation)....77
Extendable Lookup Value List....78
Extendable Lookup Value....78
Viewing the Details of the C1-CONTRACTTYPES Extendable Lookup Value....80
Editing a Value of the C1-CONTRACTTYPES Extendable Lookup....80
Deleting a Value of the C1-CONTRACTTYPES Extendable Lookup....83
Copying a Value of the C1-CONTRACTTYPES Extendable Lookup....84
Person View....86
Person Information....86
Person Tree....88
Person Characteristics....88
Person Relationship....88
Person Relationship Characteristics....89
Viewing the Person Details....90
Account View....91
Account Information....91
Account Tree....93
Account Characteristics....93
Viewing the Account Details....94
Account Collection Summary....95
Account Collection Summary - Main....95
Account Collection Summary - Collection....97
Dashboard Portal....101
Current Context Zone....101
Customer Contact Zone....102
Financial Information Zone....102
Alert Zone....104
Setting Up Bill Print Groups....106
Bill Print Group - Main....107
Bill Print Group - Contract Sub Group....107
Address....108
Prerequisites....109
Maintaining Address Status through Inbound Web Service....110
Address....111
Search Address....111
Address Entities....115
Address Characteristics....116
Searching for an Address....116
Viewing the Address Details....117
Viewing the Entities Associated to an Address....117

Copyright © 2024, Oracle and/or its affiliates.

Viewing the Characteristics of an Address.... 118
Defining an Address.... 118
Defining a Characteristic for an Address.... 120
Associating an Address to an Entity.... 121
Address Information.... 123
Address Information - Main.... 123
Address Information - Log.... 130
Start / Stop.... 132
The Big Picture Of Starting Service.... 132
The Lifecycle Of A Pending Start Contract.... 132
Activating Pending Starts.... 132
What Happens After A Contract Is Started?.... 133
The Big Picture Of Stopping Service.... 133
The Lifecycle Of A Pending Stop Contract.... 133
Finalizing Pending Stops.... 134
What Happens After A Contract Is Stopped?.... 135
Profile.... 135
Search Profile.... 135
Searching for a Profile.... 136
Defining a Profile.... 137
Defining a Characteristic for a Profile.... 138
Associating Algorithms with a Profile.... 139
Editing a Profile.... 140
Deleting a Profile.... 142
Viewing the Profile Details.... 142
Start/Stop Maintenance.... 143
Start/Stop - Main.... 143
Start/Stop - Start Confirmation.... 144
Start/Stop - Stop Confirmation.... 145
Start/Stop - Address/Phone.... 146
Start/Stop - Pending Contract.... 147
How To.... 148
The Big Picture Of Customer Contacts.... 149
Customer Contacts Are Associated With Persons, NOT Accounts.... 150
How Customer Contacts Are Created.... 150
Customer Contacts Are Used To Trigger Letters.... 150
A Customer Contact May Trigger Reminders.... 150
Customer Contacts Can Be Used As Case Files.... 151
Maintaining Customer Contacts.... 151
Customer Contact - Main.... 151
Customer Contact - Log.... 153
Customer Contact - Characteristics.... 153
Printing Letters.... 154
Letter Templates Control The Information Merged Onto Letters.... 154
BI Publisher and Crystal Reports Versus DOC 1.... 154
Where Does The Address Come From?.... 154
Technical Implementation Of Online Letter Images.... 155
Technical Implementation Of Batch Letter Production.... 155
Reproducing The Letter Print Flat File.... 156
How To Reprint A Specific Letter.... 156
The Doc 1 Records.... 156
Printer Keys (Record Types).... 156
Sort Key.... 157
Mailing Key.... 157
Letter Information.... 157
The Word Records.... 162

Copyright © 2024, Oracle and/or its affiliates.

Additional Fields For A Write Off Process / Event Letter....164
Additional Fields For A Workflow Process / Event Letter....164
Additional Fields For An Overdue Process / Event Letter....165
Maintaining Declarations....165

Chapter 2: Billing....167
The Big Picture of Billing....168
An Illustration Of A Simple Bill....168
A High Level Overview Of The Bill Creation Process....169
Bill Errors....170
Bill Segment Errors....170
Bill Completion Errors....170
Cancel / Rebill Incorrect Bill Segments....171
Credit Notes - Presenting Cancellation Details on a Separate Bill....171
Credit Notes Are Only Possible if the Installation Option Is Turned On....172
Credit / Correction Notes and Auto Pay....172
Adjustments On Credit Notes....172
Credit Note Routing Suppression....172
Correction Notes - Presenting Correction Details on a Separate Bill....173
Correction Notes Are Only Possible if the Installation Option Is Turned On....173
Correction Notes Versus Credit Notes....173
Correction Notes and Auto Pay....173
How To Override The Commodity Amount In A Bill Segment....173
How Rates Affect The Information On Bill Segments....174
Bill Frequency - Bill Cycle vs Bill Segment Duration....175
Ways To Control The Start Date Of A Bill Segment....175
Ways To Control The End Date Of A Bill Segment....175
Preventing Short Bill Segments....177
Prorating Charges When a Rate is Applied....177
Batch Billing....177
Window Billing And The Bill Cycle Schedule....177
Confirming A Batch Of Bills Before Completing Them....177
Canceling A Batch Of Bills After They're Complete....178
Reopening A Batch Of Bills After They're Complete....178
Fixing Errors Detected In Batch Billing....178
Completing Pending Bills....178
Billing Financial Transaction Considerations....178
Billing - Current Balance versus Payoff Balance....178
The Source Of GL Accounts On A Bill Segment's Financial Transaction....180
Determining the Statistical Quantity....180
The Source Of Bill Routing Information....180
Bill Messages....181
The Source Of Bill Messages....181
Substituting Field Values Into A Message....181
A Bill May Affect More Than Just Customer Balances....182
Auto Cancel / Rebill If A Contract Is Stopped Retroactively....182
Deposits May Be Refunded....182
FT Freeze Repercussions....182
Budget Billing....182
Account versus Contract....182
The Financial Impact Of Budget Plans....183
Budget Billing Background Processes....183
Canceling A Budget....184
Using Billable Charges For Pass Through / Convergent Billing....184
Printing Bills....185

Copyright © 2024, Oracle and/or its affiliates.

Bill Routings Are Created For Each Recipient Of A Bill.... 185
Bill Route Types Control The Information Merged Onto Bills.... 185
Technical Implementation Of Online Bill Images.... 185
Technical Implementation Of Printing Bills In Batch....186
Reproducing The Bill Print Flat File.... 186
How To Reprint A Specific Bill.... 186
Who Gets A Copy Of A Bill?.... 187
Final Bills and Bill Print.... 187
The Doc 1 Records....187
Writing Off Bills....197
Idiosyncratic Manual Bill Cancellation....197
Suppressing Extraction for Zero or Negligible Amount Bills.... 197
Due Date based on Bill Creation or Completion Date.... 200
Adjust Underpayments at Bill Level through Write Off....201
Default Bill Cutoff Date....202
Maintaining Bills.... 204
Bill Lifecycle.... 205
Bill - Main Information....207
Generate....210
Freeze.... 210
Cancel Frozen.... 211
Complete.... 211
Delete....212
Reopen....212
Credit Note....212
Correction Note....213
Bill - Bill Segments....213
Generate (Bill - Bill Segments)....215
Freeze (Bill - Bill Segments).... 215
Delete (Bill - Bill Segments)....216
Cancel/Rebill/Freeze (Bill - Bill Segments)....216
Cancel (Bill - Bill Segments)....217
Credit Note (Bill - Bill Segments)....217
Bill - Bill Routings....218
Bill - Bill Messages....219
Bill - Characteristics....220
Bill - Auto Pay.... 220
Stop Autopay (Bill - Auto Pay).... 221
How To....221
How To Create A Bill For All Contracts Linked To An Account.... 221
How To Create A Bill For A Specific Contract.... 221
How To Create a Bill with no Bill Segments....222
How To Create A One-Time Invoice....222
How To Correct A Bill Segment That's In Error....222
How To Correct A Bill That's In Error....223
How To Cancel A Bill Segment.... 223
How To Cancel / Rebill A Bill Segment.... 224
How To Create A Credit Note.... 224
How To Override Service Quantities On A Specific Bill Segment.... 225
How To Remove Unwanted Adjustments (or Payments) From A Completed Bill....225
How To Add An Adjustment (or Payment) To A Completed Bill....226
How To Reprint A Bill (For The Original Recipients or For Someone New).... 226
How To Add Ad Hoc Messages To A Bill....226
How To Remove A Completed Bill....226
How To Display A Bill On-line....227
How to Enable Audit Facility for Bill Reopen....227

Copyright © 2024, Oracle and/or its affiliates.

How to Re-evaluate Auto Pay Details of a Bill....227
Financial Transactions On A Bill....228
Maintaining Bill Segments....229
Bill Segment Lifecycle....229
Bill Segment - Main Information....231
Generate (Bill Segment)....232
Delete (Bill Segment)....232
Freeze (Bill Segment)....232
Rebill (Bill Segment)....233
Init Cancel (Bill Segment)....234
Undo (Bill Segment)....234
Cancel (Bill Segment)....234
Credit Note (Bill Segment)....235
Bill Segment - SQ Details....235
Bill Segment - Calc Lines....236
Bill Segment - Financial Details....237
Bill Segment - Bill Segment Messages....238
Bill Segment - Sub Bill Segments....239
Financial - Bill Segment Calculation Line Characteristics....239
Multi Cancel/Rebill....239
Multi Cancel/Rebill - Main....240
Multi Cancel/Rebill - Graph....241
Contract Billing History....241
Bill Exception....242
Bill Segment Exception....242
Invoice Frequency....242
Invoice Frequency List....242
Invoice Frequency....243
Viewing the Invoice Frequency Details....243
Defining an Invoice Frequency....244
Editing an Invoice Frequency....245
Deleting an Invoice Frequency....246
Charge Type....246
Charge Type List....246
Charge Type....247
Defining a Charge Type....248
Defining a Characteristic for a Charge Type....249
Editing a Charge Type....250
Copying a Charge Type....251
Deleting a Charge Type....252
Viewing the Charge Type Details....253
Maintaining Billable Charges....254
Billable Charge - Main....254
Billable Charge - Line Characteristics....256
Billable Charge - SQ Details....256
Billable Charge - Characteristics....257
Billable Charge - Pricing Information....257
Uploading Billable Charges....258
Billable Charge Upload Background Processes....258
Process X - Populate BC Upload Staging....259
BCU1 - Validate & Populate Billable Charge Upload Staging....264
BCU2 - Create Billable Charge....264
BCUP-PRG - Purge Billable Charge Upload Objects....265
Billable Charge Upload Staging....265
Billable Charge Upload - Main....265
Billable Charge Upload - Lines....266

Copyright © 2024, Oracle and/or its affiliates.

Billable Charge Upload - SQ Details....266
Billable Charge Upload - Characteristics....266
Billable Charge Upload Exception....266
Budget Review....266
Bill Segment Summary....267
Search....267
Searching for a Bill Segment....270
Bill Drill Down Details....271
Search Billing Element....271
Bill Segments....275
SQ Details....276
Bill Segment Calculation Lines....277
Timeline and Billable Charge Details....277
Audit Event Details....278
Pricing Calculation Details....279
Transaction Details....279
Transaction Leg Calculation Lines....280
Searching for a Bill....281
Tracing the Source of a Bill (Non-TFM)....281
Tracing the Source of a Bill Generated Through TFM....282
Tracing the Source of a Self-Funded Bill....283
Tracing the Source of a Fully-Insured Group Bill....284
Viewing the Bill Segments of a Bill....284
Viewing the SQ Details of a Bill Segment....285
Viewing the Calculation Lines of a Bill Segment....285
Viewing the Timeline and Billable Charge Details of a Bill Segment....286
Viewing the Audit Event Details of a Timeline....286
Viewing the Pricing Calculation Details of a Timeline....287
Viewing the Transactions for Which the Bill Segment is Created....288
Viewing the Transaction Leg Calculation Lines of the Transaction....288
Tracing the Source of a Billable Charge Generated Through TFM....289
Tracing the Source of a Self-Funded Billable Charge....290
Tracing the Source of a Fully-Insured Group Billable Charge....290
Viewing the Bill Segments of a Billable Charge....290
Alert Notification for Bill and Bill Segment Errors....291

Chapter 3: Payments....293
The Big Picture of Payments....294
A Payment Event Has Payments And Tenders....294
Multiple Tenders Used To Pay For Multiple Accounts....294
An Overview Of The Payment Event Creation & Allocation Process....295
Distributing A Payment Event....296
Distributing A Payment Amongst An Account's Contracts....296
Overpayment....298
Canceling A Tender Versus Canceling A Payment....298
NSF Cancellations....299
Transferring A Payment....299
Unbalanced Payment Events....299
How And When Payments Appear On Bills....300
Tender Management and Workstation Cashiering....300
Managing Your Cash Drawers....301
Turn Ins....303
Balancing By Tender Type....303
Cash Back....303
Managing Payments Interfaced From External Sources....304

Copyright © 2024, Oracle and/or its affiliates.

Exceptions....304
Payment Exceptions....304
Payment Event Exceptions....305
Resolving Exceptions Automatically....305
Payment Financial Transaction Considerations....305
Payment - Current Balance versus Payoff Balance....305
The Source Of GL Accounts On A Payment Financial Transaction....305
A Payment May Affect More Than Just Customer Balances....306
Open Item Accounting and Match Events....306
FT Freeze Repercussions....306
Automatic Payments....306
How To Set Up A Customer To Pay Automatically....306
What Are Automatic Payments?....306
How And When Are Automatic Payments Created?....306
Automatic Payment Dates....308
How To Implement Maximum Withdrawal Limits....308
How Are Automatic Payments Cancelled?....308
Match Events Are Created For Open-Item Customers When An Automatic Payment Is Created....309
Promise To Pay and Automatic Payments....309
Non-billed Budgets and Automatic Payment....309
Downloading Automatic Payments and Interfacing Them To The GL....309
How To Split Automatic Payment....310
Rule Based Auto Pay Instructions....311
ACH Record Layouts....313
Issuing A Payment Advice Instead Of Creating An Automatic Payment....316
How To Set Up A Customer To Receive Payment Advices....316
Payment Advice Option Is For Bill-Related Automatic Payments Only....316
Partial Payment Transfer....316
Partial Payment Transfer through Payment Request....316
Partial Payment Transfer through Mass Payment Transfer....320
Partial Payment Transfer through Inbound Web Service....324
Three-Step PUPL Processing....327
Handling Multiple Payment Records of the Same Payee Account through Three-Step PUPL Batch....328
Maintaining Payment Events....348
Payment Lifecycles....348
Payment Event Lifecycle....348
Tender Lifecycle....349
Payment Lifecycle....349
Payment Event - Add Dialog....350
Payment Event - Main Information....352
Payment Event - Tenders....353
Payment Event Action Codes....355
Payment Actions....355
Payment Event Actions....356
Tender Actions....358
Payment Event QuickAdd....358
Multiple Payment Events Dialog....359
Single Payment Event Dialog....360
Payment Quick Add....361
Maintaining Payments....363
Payment - Main....363
Payment - Pay Segments....365
Payment - Manual Distribution....366
Payment - Characteristics....367

Copyright © 2024, Oracle and/or its affiliates.

Payment Action Codes.... 367
Distribute (A Payment).... 367
Redistribute (A Payment).... 367
Freeze (A Payment).... 368
Cancel (A Payment).... 368
Transfer (A Payment).... 368
Delete (A Payment).... 369
Payment Portal.... 369
Credit Card Payments.... 369
Non CIS Payments.... 370
How To.... 370
How To Add A New Payment Event.... 370
How To Cash A Check.... 370
How To Allocate The Tender Amount To Multiple Accounts.... 371
How To Print Receipts And Endorsements.... 371
How To Cancel A Tender.... 371
How To Transfer A Payment From One Account To Another.... 372
How To Distribute A Payment To A Specific Contract.... 372
How To Get An Unbalanced Tender Control In Balance (Fixing Over/Under).... 372
How To Add A Non-CIS Payment Using A Template.... 373
How To Authorize A Credit Card Tender.... 374
How To Reverse A Credit Card Tender.... 374
How to Distribute Overpayment on the Next Bill Completion.... 375
Financial Transactions On A Payment.... 375
Payment History.... 376
Search.... 376
Searching for a Payment.... 378
Remittance Summary.... 379
Search.... 379
Payments.... 380
Searching for Payments Remitted by a Bank.... 381
Account Payment History.... 382
Payment Tender History.... 382
Payment / Tender Search.... 382
Payment Event Exception.... 385
Payment Exception.... 386
Maintaining Deposit Controls.... 386
The Lifecycle Of A Deposit Control.... 386
Deposit Control - Main.... 387
Deposit Control - Tender Control.... 387
Deposit Control - Tender Deposit.... 388
Deposit Control - Turn Ins.... 388
Maintaining Tender Controls.... 388
The Lifecycle Of A Tender Control.... 389
Tender Control - Main.... 389
Tender Control - Tenders.... 391
Tender Control - Turn Ins.... 391
Tender Control - Exceptions.... 392
Tender Control - Characteristics.... 392
Interfacing Payments From External Sources.... 392
Interfacing Payments.... 392
Populating The Payment Upload Staging Records.... 392
PYUP-PRG - Purge Payment Upload Objects.... 400
Maintaining Deposit Control Staging.... 400
Payment Upload Staging.... 402
Payment Upload Exception.... 404

Copyright © 2024, Oracle and/or its affiliates.

Interfacing Payments Using Distribution Rules....404
Populating The Payment Event Upload Staging Records....404
Payment Event Upload Staging....412
Payment Creation, Cancellation, Transfer, and View Through Inbound Web Service....413
Prerequisites....415
C1-CreatePayment Business Service....415
Payment Creation Request in XML Format....416
Payment Creation Response in XML Format....416
Payment Creation Request in JSON Format....417
Payment Creation Response in JSON Format....418
Payment Creation Request and Response - Tags in XML Format....419
Payment Creation Request and Response - Attribute/Value Pairs in JSON Format....423
Sample Payment Creation Request and Response in XML Format....426
Sample Payment Creation Request and Response in JSON Format....428
C1-CancelPayment Business Service....431
Payment Cancellation Request in XML Format....431
Payment Cancellation Response in XML Format....431
Payment Cancellation Request in JSON Format....432
Payment Cancellation Response in JSON Format....432
Payment Cancellation Request and Response - Tags in XML Format....432
Payment Cancellation Request and Response - Tags in JSON Format....434
Sample Payment Cancellation Request and Response in XML Format....435
Sample Payment Cancellation Request and Response in JSON Format....436
C1-TransferPayment Business Service....437
Payment Transfer Request in XML Format....437
Payment Transfer Response in XML Format....437
Payment Transfer Request in JSON Format....438
Payment Transfer Response in JSON Format....438
Payment Transfer Request and Response - Tags in XML Format....438
Payment Transfer Request and Response - Attribute/Value Pairs in JSON Format....440
Sample Payment Transfer Request and Response in XML Format....442
Sample Payment Transfer Request and Response in JSON Format....442
C1-ViewPayment Business Service....443
Payment View Request in XML Format....444
Payment View Response in XML Format....444
Payment View Request in JSON Format....444
Payment View Response in JSON Format....445
Payment View Request and Response - Tags in XML Format....445
Payment View Request and Response - Attribute/Value Pairs in JSON Format....447
Sample Payment View Request and Response in XML Format....449
Sample Payment View Request and Response in JSON Format....450

Chapter 4: Adjustments....453
The Big Picture Of Adjustments....454
Adjustments - Current Balance versus Payoff Balance....454
When Current Balance Equals Payoff Balance....454
When Current Balance Differs From Payoff Balance....454
Adjustment Type And Balances....455
Canceling Adjustments....456
Transfer Adjustments....456
Calculated Adjustments....456
Adjustment Amount May Be Positive, Negative Or Zero....456
Adjustment Type Controls Everything....457
Controls Which Balance(s) Are Affected....457
Defines The GL Account Affected By The Adjustment....457

Copyright © 2024, Oracle and/or its affiliates.

Defaults The Adjustment Amount.... 457
Calculates The Adjustment Amount.... 458
Controls The Interface To A/P & 1099 Reporting....458
Controls Information Printed On The Bill.... 458
Controls If The Adjustment Can Be Frozen Prior To Bill Completion....458
Controls If The Adjustment Requires Approval.... 458
Unbilled Adjustments And Aged Debt.... 458
An Adjustment May Affect More Than Just Customer Balances....459
Maintaining Adjustments....459
The Lifecycle Of An Adjustment....459
Adjustments - Main Information....461
Adjustments - Characteristics....463
Adjustments - Transfer Adjustment.... 463
Adjustments - A/P Request.... 464
Adjustments - Approval....465
Financial - Adjustment Calculation Line Characteristics....465
How and When To Use An Adjustment....466
How To Create A Transfer Adjustment.... 466
How To Create A Calculated Adjustment.... 466
How To Cancel An A/P Adjustment After It Has Been Selected By A/P....467
How To Correct Normal Contract Debt....467
How To Apply Adhoc Fees To A Contract.... 467
How To Apply Interest To A Deposit....467
How To Apply A Deposit To Outstanding Debt.... 468
How To Refund A Deposit With A Check....468
How To Write-Down Deposit Due....468
How To Refund A Credit Balance With A Check....468
How To Transfer A Credit Balance To A Contract In Arrears....469
How To Change The Age Of Debt....469
How To Write-Down A Charitable Contribution Contract Debt.... 469
How To Write-Off Debt.... 469
How To Cancel A Contract's Budget....470
How To Adjust A Contract On A Budget.... 470
How To Adjust An Installment Plan Contract....470
How To Use An Adjustment To Change The GL Distribution.... 470
How to Create an Adjustment against a Completed Bill....471
Interfacing Adjustments From External Sources.... 471
Interfacing Adjustments....471
Process X - Populate Adjustment Upload Records....472
Process C1-ADUP1 - Preprocess Adjustment Uploads.... 476
Process C1-ADUP2 - Upload Adjustments.... 477
Suspense Adjustments.... 478
What Are Suspense Adjustments?.... 478
How Are Suspense Adjustments Resolved?.... 478
Maintaining Adjustment Staging Control.... 479
Maintaining Adjustment Upload Staging....479
Adjustment Creation, Cancellation, and View Through Inbound Web Service....480
Prerequisites.... 480
C1-CreateAdjustment Business Service.... 482
Adjustment Creation Request in the XML Format....483
Adjustment Creation Response in the XML Format.... 483
Adjustment Creation Request in the JSON Format.... 484
Adjustment Creation Response in the JSON Format....485
Adjustment Creation Request and Response - Tags in XML Format.... 486
Adjustment Creation Request and Response - Attribute/Value Pairs in JSON Format.... 491
Sample Adjustment Creation Request and Response in the XML Format.... 496

Copyright © 2024, Oracle and/or its affiliates.

Sample Adjustment Creation Request and Response in the JSON Format....497
C1-CancelAdjustment Business Service....498
Adjustment Cancellation Request in the XML Format....499
Adjustment Cancellation Response in the XML Format....500
Adjustment Cancellation Request in the JSON Format....500
Adjustment Cancellation Response in the JSON Format....501
Adjustment Cancellation Request and Response - Tags in XML Format....501
Adjustment Cancellation Request and Response - Attribute/Value Pairs in JSON Format....503
Sample Adjustment Cancellation Request and Response in the XML Format....505
Sample Adjustment Cancellation Request and Response in the JSON Format....505
C1-GetAdjustmentHistory Business Service....506
Adjustment View Request in the XML Format....507
Adjustment View Response in the XML Format....508
Adjustment View Request in the JSON Format....509
Adjustment View Response in the JSON Format....509
Adjustment View Request and Response - Tags in XML Format....510
Adjustment View Request and Response - Attribute/Value Pairs in JSON Format....513
Sample Adjustment View Request and Response in the XML Format....516
Sample Adjustment View Request and Response in the JSON Format....518

Chapter 5: Credit & Collections....521
The Big Picture Of Credit & Collections....522
Automating Your C&C Activities....522
The Lifecycle Of A Collection Process And Its Events....522
What Is A Collection Process?....522
How Are Collection Processes Created?....522
How Are Collection Events Completed?....523
How Are Collection Processes (and their Events) Canceled?....523
The Collection Process Background Processes....523
Account Debt Monitor....524
Collection Event Activator....524
Collection Process Monitor....524
The Lifecycle Of A Write Off Process And Its Events....524
What Is A Write-off Process?....525
How Are Write-off Processes Created?....525
How Are Write-off Events Completed?....526
How Are Write-off Processes (and their Events) Canceled?....526
What Happens If The Debt Is Not Paid?....526
How Is Debt Referred To Collection Agencies....526
The Write-Off Process Background Processes....526
Write-Off Monitor....527
Write-Off Event Activator....527
Collection Process Maintenance....527
Collection Process - Main Information....527
Collection Process - Contracts....529
Collection Process - Events....530
How To Perform Common Collection Process Functions....531
How To Create A Collection Process....531
How To Change Collection Events....532
How To Cancel A Collection Process....532
Write-off Process Maintenance....533
Write-off Process - Main Information....533
Write-off Process - Contracts....534
Write-off Process - Events....535
How To Perform Common Write-off Process Functions....536

Copyright © 2024, Oracle and/or its affiliates.

How To Create A Write-off Process....536
How To Change Write Off Events....537
How To Cancel A Write Off Process....537
Writing Off Uncollectable Debt....537
Write Off - Main....538
How To Manually Write-Off Debt....538
Write Off - Write Off Contracts....538
Collection Referral....538
Payment Arrangements....539
Setting Up Payment Arrangements....539
Pay Arrangement - Main....539
How To Set Up A Payment Arrangement....541
How To Add Additional Debt To A Payment Arrangement....541
How To Break A Payment Arrangement....541
How To Cancel A Payment Arrangement....542
Pay Arrangement - History....542
Promise To Pay....543
Search....544
Searching a Promise To Pay....545
Adding a Promise To Pay....545
Editing Promise To Pay....546
Cancelling Promise To Pay....548

Chapter 6: Financial Transactions....549
The Big Picture Of Financial Transactions....550
Bill Segment Financial Transactions....550
Payment Segment Financial Transactions....550
Adjustment Financial Transactions....550
Financial Transactions And Aged Debt....550
Current Balance versus Payoff Balance....551
Adjustments - Current Balance versus Payoff Balance....551
Billing - Current Balance versus Payoff Balance....551
Payment - Current Balance versus Payoff Balance....551
The Source Of GL Accounts On Financial Transactions....551
Obscure Things That Can Happen....551
A Stopped Contract May Be Closed....551
A Closed Contract May Be Reactivated....551
A Write-Off Process May Be Deactivated If The Contract Is Closed....552
A Reactivated Contract May Be Closed....552
One Or More Algorithms May Be Executed....552
The Big Picture of Balance Control....552
The Balance Control Background Processes....552
Balance Control Information Is Available Online....554
The GL Interface....554
GLASSIGN - Assign GL Account Numbers To GL Details....554
GLS - Prepare FTs for Download....554
GLDL - Create General Ledger Download Flat File....555
GL Posting for Zero Valued Transaction....556
Different GL Accounting Currency....556
Financial Transaction....557
Financial Transaction - Main....557
Financial Transaction - FT Process....559
Account Financial History....560
Account Bill/Payment History....561
Contract Financial History....562

Copyright © 2024, Oracle and/or its affiliates.

Contract Cash Accounting Balance....562
Balance Control....563
Match Event....564
Match Event - Main....564
Match Event - FT Details....566
Match Event - Subtotals....568
How To Perform Common Match Event Functions....569
How To Find The Match Event Associated With A Financial Transaction....569
How To Dispute An Item....569
How To Match A Small Mismatch....569
Billed Revenue....570
Viewing Billed Revenue....570
FT GL Drill Down Details....571
Search GL Extract File....572
Searching for a GL Extract File....573
GL Extract File Details....573
Financial Transaction Details....574
Bill Segment Details....575
Adjustment Details....576
Payment Details....576
Viewing the Details of a GL Extract File....577
Viewing Financial Transactions Posted in a GL Account....577
Viewing the Financial Transaction Details....577

Chapter 7: Statements....579
The Big Picture of Complex Statements....580
Constructing Statements....581
Producing Statements....581
Statement Formats....581
Statement Cycles....642
Create Statements Background Process....642
Statement Routing....643
Where Are Statements Sent....643
Statement Generation Process....643
On-line Statement Production....645
How to Generate a Statement....645
How to Regenerate a Statement....645
How to Search for a Statement....646
How to View a Statement....646
How to Download a Statement....646
How to Archive a Statement....646
C1_RegenerateStatement Inbound Web Service Schema....647
C1_SearchStatement Inbound Web Service Schema....647
C1_ReprintStatement Inbound Web Service Schema....647
XML Format Required for C1_RegenerateStatement....647
XML Format Required for C1_SearchStatement....648
XML Format Required for C1_ReprintStatement....649
Printing Statements....649
Statement Route Types Control The Information Merged Onto Statements....649
Technical Implementation Of Online Statement Image....650
Technical Implementation Of Printing Statements In Batch....650
Reproducing The Statement Print Flat File....650
How To Reprint A Specific Statement....650
The Doc 1 Records....650
Statement Construct Maintenance....656

Copyright © 2024, Oracle and/or its affiliates.

Lifecycle of a Statement Construct....656
Statement Construct - Main....657
Statement Construct - Details....657
Statement Construct - Characteristics....658
Statement Maintenance....659
Lifecycle of a Statement....659
Statement - Main....659
Statement - Details....660

Chapter 8: Sales & Marketing....663
The Big Picture of Campaigns, Packages and Orders....664
An Overview of Campaigns, Packages and Orders....664
Supported Business Processes....664
Marketing to Prospects from a Marketing List....665
Marketing to Pre-selected Customers....666
Real-time Marketing of Additional Services to a Customer....666
Real-time Marketing of Services to a Prospect....667
Setting up a New Customer Prior To Using Start/Stop....667
Examples of Campaigns and Packages....669
Campaigns Without Packages....669
Campaigns With Simple Packages....672
The Big Picture Of Package Eligibility Rules....674
Criteria Groups versus Eligibility Criteria....674
Defining Logical Criteria....675
Examples Of Package Eligibility Rules....676
A Package That Is Always Eligible....676
A Package With More Complex Operators....676
The Big Picture Of Campaign Eligibility Rules....677
Campaign Eligibility Rules Are Not Strictly Enforced....677
Campaign Eligibility Rules Are Defined Using Algorithms....677
Examples Of Campaign Eligibility Rules....678
A Campaign That Is Always Eligible....678
A Campaign That Is Never Eligible....678
Designing Campaigns and Packages....678
Determine The Place Of Start / Stop....678
Extract Algorithms From The Eligibility Rules....679
Extract Miscellaneous Fields From The Eligibility Rules....679
Determine Additional Information That Is Required If A Package Is Selected....679
Determine If There Are Additional Questions / Fields For Your Packages....679
Determine If Additional Instructions Should Be Shown....680
Determine If There Are Campaign-Level Fields / Questions....680
Determine The Properties Of Every Miscellaneous Field....681
Extract Characteristic From The Miscellaneous Fields....682
Extract Column References From The Miscellaneous Fields....683
Extract Algorithms From The Column References....684
Determine Account Dependency For Each Column Reference....685
Determine If Any Properties Are Overridden On Any Package....685
Determine The Eligibility Rules For The Campaign....685
Determine How The Campaign Behaves....686
Setup Sequence....686
Other Useful Information....687
Order versus Start / Stop....687
Orders Cannot Be Changed After They Are Completed....688
To Do List Processing....688
Hold and Cancel Reason Codes....688

Copyright © 2024, Oracle and/or its affiliates.

Additional Things Can Happen When An Order Is Completed.... 688
New Account is Created if an Order is Completed.... 688
Account Usage is Controlled by the Campaign.... 688
Maintaining Orders.... 688
Lifecycle of an Order.... 689
Order User Interface Flow.... 689
Order - Main.... 690
Order - Main - Person Information.... 691
Order - Main - Account Information.... 693
Order - Questions & Misc. Fields.... 694
Order - Log.... 696
Order - Package Confirmation.... 697
Order - Hold Dialog.... 697
Order - Cancel Dialog.... 698
Maintaining Campaigns.... 698
Lifecycle of a Campaign.... 698
Campaign - Main.... 699
Campaign - Questions and Misc. Fields.... 700
Campaign - Algorithm.... 701
Campaign - Package.... 702
Maintaining Packages.... 702
Lifecycle of a Package.... 703
Package - Main.... 703
Package - Contracts To Create.... 703
Package - Questions & Misc Fields.... 704
Package - Eligibility.... 705
Package - Instructions.... 706

Chapter 9: Rates.... 707
All Rates Share A Common Structure.... 708
How To Create A New Rate.... 708
Control Tables That Must Be Set Up Before Creating A Rate Schedule.... 709
Defining Frequency Codes.... 709
UOM versus Variance Parameter versus SQI.... 709
Setting Up Register Rules.... 710
Base Package Register Rules.... 711
Setting Up Service Quantity (SQ) Rules.... 712
Base Package SQ Rules.... 713
Defining Service Quantity (SQ) Identifiers.... 720
Setting Up A Rate Schedule.... 720
Rate Schedule - Main.... 721
Rate Schedule - SQ Rules.... 722
Rate Schedule - Register Rules.... 722
Rate Schedule - Bill Messages.... 722
Rate Schedule Merge.... 723
Removing A Row.... 724
Adding A New Row.... 724
Removing An Uncommitted Row.... 724
Moving Rows Up and Down.... 724
Setting Up Bill Factors.... 725
An Overview Of Bill Factors.... 725
The Structure Of A Bill Factor.... 726
An Illustration Of A Bill Factor And Its Characteristics.... 726
Deriving / Passing In Characteristic Values.... 727
Bill Factor Interval Values.... 728

Copyright © 2024, Oracle and/or its affiliates.

Interval Values Can Be Used For More Than Just Interval Prices....728
Interval Data is Stored in Data Sets....729
Bill Factor Variance Parameter Values....729
Variance Parameter Values Can Be Used For More Than Just Prices....730
Variance Parameter Values May Differ By Customer....730
Defining Bill Factors....730
Defining General Bill Factor Information....730
Setting Up Bill Factor Characteristics....732
Defining Bill Factor Values....733
Bill Factor Value - Main....733
Bill Factor Value - GL Distribution....734
Defining Interval Values....734
Bill Factor Interval Values Query....735
Upload Interval Values....735
Process X - Populate Interval Data....736
Interval Value Set....736
Interval Value....737
Interval Value Set Key....737
Defining Bill Factor Variance Parameter Values....737
Defining Rate Versions....738
Lifecycle of a Rate Version....738
Rate Version - Main....739
Rate Version - Bill Print Info....740
How To Use Description on Bill....740
Designing Rate Components....743
Rate Component Design Methodology....743
Rate Component Rounding....746
Rounding Precision Is Defined On A Rate Component....746
Rounding Method Is Defined On A Rate Component....747
Rounding And FCPO Rate Components....747
Interim Rounding For Apply To Rate Components....747
Rounding At The End....748
The Big Picture Of Rate Component Eligibility Rules....748
A Rate Component Is Eligible By Default....749
Criteria Groups versus Eligibility Criteria....749
Defining Logical Criteria....750
Examples Of Rate Component Eligibility Rules....750
Defining Rate Components....753
Rate Component - Main Information....753
How To....755
Rate Component - Cross Reference....759
Rate Component - GL Distribution....759
Rate Component - Characteristics....760
Rate Component - Eligibility....760
Rate Version Merge....761
Resequencing Rate Components....763
Adding A New Row To A Rate....763
Removing An Uncommitted Row From A Rate....763
Moving Positional Rows Up and Down....763
Rate Component Cross Reference Information....763
Rate Check....765
Rate Check - Main....765
Rate Check Results - Calc Lines....766
Rate Check Results - SQ Details....767
Rate Check Results - Characteristics....767
Rate Schedule Summary....767

Copyright © 2024, Oracle and/or its affiliates.

Search....767
Rate Components....768
Searching for a Rate Schedule....769
Viewing Rate Components of a Rate Schedule....769
Effective Dates & Price Proration....770
How To Add A New Service Quantity Rule Algorithm....770
How To Add A New Register Rule Algorithm....771

Chapter 10: Quotes....773
The Big Picture Of Quotations....774
Basic Quotation Concepts....774
A Quote Shows How Bill Segments Might Look....774
Proposal Contracts Must Exist....774
Creating Quotes And Quote Details....777
Completing A Quote Creates Quote Routings (And More)....778
Accepting / Declining Quote Details....778
Tinkering With Proposal Contracts After Quote Generation....779
Closing Quotes....779
Alerts....779
Printing Quotes....779
Quote Routings Are Created For Each Recipient Of A Quote....779
Quote Route Types Control The Information Merged Onto Quotes....779
Technical Implementation Of Online Quote Image....780
Technical Implementation Of Printing Quotes In Batch....780
Reproducing The Quote Print Flat File....780
How To Reprint A Quote (For The Original Recipients or For Someone New)....780
The Doc 1 Records....781
Maintaining Quotes....787
The Lifecycle Of A Quote And Quote Detail....787
Quote Lifecycle....787
Quote Detail Lifecycle....788
Quote - Main....788
Action Buttons On Quote Main....789
Quote - Quote Details....791
Quote - Quote Routings....792

Chapter 11: Service Credits....795
The Big Picture Of Service Credit Membership....796
Who are the Members?....796
Joint Membership....796
Changing the Membership Accounts....796
Membership Application....797
Creation of a Membership....797
Membership Fees....797
Stopping a Membership vs. Inactivating a Membership....797
The Big Picture of SC Events....798
Event Amounts May Contribute to a Balance....798
Events May Indicate a Fiscal Year....799
Events May Indicate a Subcategory....799
An Event May Cause Other Actions to Occur....799
Financial Transactions for an Event....800
How Events Are Created....800
Events Created by an Algorithm....800
Events Created as a Result of a Background Process....800

Copyright © 2024, Oracle and/or its affiliates.

Creating Pending Events.... 800
Events May Be Transferred....801
CPCRALOC - Allocating Capital Credits.... 801
CPCRRETR - Capital Credit Retirement....802
Maintaining Service Credit Memberships and Their Events.... 803
Service Credit Membership.... 803
Lifecycle of a Membership.... 803
Service Credit Membership - Main....804
Service Credit Membership - Account....804
Service Credit Membership - Characteristics....805
Service Credit Membership - Membership Tree....805
Maintaining Service Credit Events....805
Lifecycle of a Service Credit Event....805
Service Credit Event - Main.... 806

Chapter 12: Loans....809
The Big Picture Of Loans.... 810
Loans Are Contracts.... 810
Loan Interest Charges....810
Loan Amortization Schedules.... 810
Negative Amortization....811
Loan Overpayments....811
The Lifecycle Of A Loan....811
Finalizing And Closing A Loan....812
Navigating To The Loan Maintenance Page....812
Loan Maintenance....812
How To....814
How To Start A Loan Contract....814
How To Modify The Terms Of A Loan.... 814
How To Adjust Loans.... 815
How To Stop A Loan Contract.... 815

Chapter 13: Deposits....817
The Big Picture Of Deposits....818
Cash Deposits....818
Refunding Deposits....818
Partial Refunds....819
Total Amount To Bill.... 819
Current Balance versus Payoff Balance.... 820
Multiple Deposits Linked To A Single Account.... 821
Deposits and Credit & Collections....821
Deposit Seizures When Other Contracts Are Stopped....821
Non-Cash Deposits.... 822
3 rd Party Deposits....822
Deposit Class Controls Everything.... 823
Controls Interest Calculation.... 823
Controls The Recommended Deposit Amount.... 823
Defines The Conditions That Cause The System To Recommend A New or Additional Deposit....823
Defines Automatic Refund Conditions.... 824
Controls How Deposits Are Refunded To A Customer....824
Deposit Background Processes....824
Deposit Interest.... 824
Deposit Refund.... 824

Copyright © 2024, Oracle and/or its affiliates.

Review Deposits.... 824
Deposit Review.... 825

Chapter 14: Case Management.... 827
Background Topics.... 828
Case Type.... 828
Creating Cases.... 828
Log Information.... 828
Alerts.... 828
To Dos and Cases.... 828
Reports.... 829
Maintaining Cases.... 829
Case - Main.... 829
Case - Case Portal.... 830
Case - Log.... 831

Chapter 15: Job Streams.... 833
Job Stream Summary.... 834
Job Stream Details.... 834

Chapter 16: Workflow and Notifications.... 837
The Lifecycle Of A Workflow Process And Its Events.... 838
What Is A Workflow Process?.... 838
A Workflow Process Contains Relevant Data.... 839
How Are Workflow Processes Created?.... 840
How Are Workflow Events Completed?.... 840
How Are Workflow Processes (and their Events) Canceled?.... 841
Workflow Process Maintenance.... 841
Workflow Process - Main.... 841
Workflow Process - Char/Context.... 842
Workflow Process - Events.... 842
Workflow Process - Event Char/Context.... 844
Workflow Event Exception.... 844
Uploading Notifications.... 844
Notification Upload Background Process.... 844
Process X - Populate Notification Upload Staging.... 845
Process WFPRINIT - Initiate Workflow Processes To Execute Notification Upload Staging Records.... 848
Validation of Data Related to the Notification Upload Staging Records.... 849
Notification Upload Staging Maintenance.... 849
Notification Upload Staging - Main.... 849
Notification Upload Staging - Characteristics/Context.... 850
Notification Upload Staging - Sample Upload Extension Record.... 850
Notification Upload Exception.... 851
Downloading Notifications.... 851
Notification Download Background Processes.... 851
Process NDSXTR - Interface Notification Download Records (out of the system).... 851
Notification and XAI.... 852
NDS Contains Interesting Data.... 852
Notification Download Staging.... 852
Notification Download Staging - Main.... 852
Notification Download Staging - Characteristics/Context.... 853
Notification Download Staging - XAI Download Staging.... 853

Copyright © 2024, Oracle and/or its affiliates.

Notification Download Exception.....853

Chapter 17: Overdue Financial Obligations.....855
Overdue Processing Background Information.....856
Overdue Process Maintenance.....856
Overdue Process - Main.....856
Overdue Process - Events.....857
Overdue Process - Log.....858
Overdue Process - Characteristics.....858
How To Perform Common Overdue Process Functions.....859
How To Create An Overdue Process.....859
How To Change Overdue Events.....859
How To Cancel An Overdue Process.....860
Payment Arrangement for Bills.....860
How To Set Up A Payment Arrangement.....862
How To Add Additional Debt To A Payment Arrangement.....862
How To Break A Payment Arrangement.....862
How To Cancel A Payment Arrangement.....863

Chapter 18: Data Extractor.....865
Data Extraction.....866
Transactional Business Objects.....866
Master Data Business Objects.....868
Modes of Data Extraction.....868
Prerequisites.....869
Creating the DTEXSCH Feature Configuration.....869
Data Extract Request Type.....870
Search Data Extract Request Type.....870
Data Extract Request Type.....871
Searching for a Data Extract Request Type.....871
Defining a Data Extract Request Type.....872
Editing a Data Extract Request Type.....876
Deleting a Data Extract Request Type.....880
Viewing the Data Extract Request Type Details.....880
Data Extract Request Dashboard.....881
Search Data Extract Request.....881
Searching for a Data Extract Request.....882

Chapter 19: File Upload Interface.....885
File Upload Process.....886
Decrypting a File.....886
Upload and Transforming a File.....888
Processing a File Request.....893
Archiving a File and/or File Records.....897
Polling Files on SFTP Server.....897
Prerequisites.....897
Configuring Data Transformation in a File Request Types.....898
Mapping Header Details for Transformation.....898
Mapping Footer Details for Transformation.....898
Mapping Fields for Data Transformation.....898
Setting Default Values for a Field.....898
Setting Mandatory Attribute of a Field.....899
Setting a Date/Time Format for a Field.....899

Copyright © 2024, Oracle and/or its affiliates.

Setting Record Identifier for Grouping Duplicate Records.... 899
Skipping BO Level Validations for a Field.... 899
Configuring Static Header and Footer for a File.... 899
Supported Functions for Default Value.... 900
Fetch Functions.... 900
Copy Functions.... 924
String Handling Functions.... 927
Script Functions.... 934
EVAL function.... 935
DUPLICATE function.... 936
Config Function.... 936
Header Function.... 936
TO DO Generation.... 937
Concat Function.... 938
Configuring the Approval Workflow Process for File Upload Interface.... 939
Approval Workflow Group.... 939
Approval Workflow Chain.... 940
Approval Workflow Criterion Type.... 940
Approval Workflow Group Chain Linkage.... 940
Approval Workflow Settings.... 941
Configuring Rules Engine for Polling Files on SFTP Server.... 941
Rule Types.... 941
Rules.... 942
Setting the C1-FLUPLD Feature Configuration.... 942
Setting the File Upload Interface Master Configuration.... 946
Setting the Email Sender Master Configuration.... 950
Setting the File Management System Configuration.... 951
File Management System.... 952
Search File or Directory.... 953
Searching for a File or a Directory.... 957
Viewing the Files of a Directory.... 957
Downloading a File from the SFTP Server.... 958
Uploading a File to the SFTP Server.... 958
File Request Type.... 959
File Request Type List.... 960
File Request Type.... 960
Defining a File Request Type.... 966
Adding Services in a File Request Type.... 974
Adding Messages in a File Request Type.... 975
Adding Header Data Transformation Details in a File Request Type.... 976
Adding Footer Data Transformation Details in a File Request Type.... 978
Adding Field Data Transformation Details in a File Request Type.... 979
Editing a File Request Type.... 981
Deleting a File Request Type.... 989
Copying a File Request Type.... 989
Viewing the File Request Type Details.... 997
File Upload Dashboard.... 997
Search File.... 998
File Upload Failure Reason.... 1000
Successfully Uploaded Record Summary.... 1001
File Record Status Summary.... 1002
File Error Messages List.... 1002
File Header Detail.... 1003
File Footer Detail.... 1003
Searching for File.... 1004
Marking a File for Archival.... 1004

Copyright © 2024, Oracle and/or its affiliates.

Viewing the Details of a File.....1005
File Record Details.....1005
File Records.....1006
File Record Service Details.....1008
File Record Identifier Details.....1008
File Record Status Transition.....1009
File Record Error Message History.....1009
Filtering a File Record.....1010
Viewing the Details of a File Record.....1010
Updating the Status of a File Record.....1011
Marking a File Record for Archival.....1011
Viewing the Record Payload.....1012
Viewing the Service Payload.....1012
Viewing the Transformed Payload.....1013

Chapter 20: Extract Template.....1015
Extract Template.....1017
Extract Template List.....1017
Extract Template.....1018
Defining an Extract Template.....1021
Defining the File Formats for an Extract Template.....1025
Adding Business Services or Business Objects in an Extract Template.....1027
Defining the Record Count for a List Element.....1028
Mapping Fields in an Extract Template.....1029
Editing an Extract Template.....1030
Copying an Extract Template.....1035
Deleting an Extract Template.....1039
Viewing the Details of an Extract Template.....1039

Chapter 21: Batch Scheduler.....1041
Prerequisites.....1042
Oracle Scheduler Interface Architecture.....1042
DBMS Import/Export.....1042
Exporting the DBMS Scheduler Data.....1043
Importing the DBMS Scheduler Data.....1044
Scheduler Chain.....1045
Search Scheduler Chain.....1045
Searching for a Scheduler Chain.....1046
Defining a Scheduler Chain.....1046
Viewing a Scheduler Chain Details.....1048
Editing a Scheduler Chain.....1049
Deleting a Scheduler Chain.....1050
Scheduler Job.....1051
Search Scheduler Job.....1051
Searching for a Scheduler Job.....1052
Defining a Scheduler Job.....1053
Viewing a Scheduler Job Details.....1055
Editing a Scheduler Job.....1055
Deleting a Scheduler Job.....1056
Submitting a Scheduler Job.....1057
Scheduler Job Monitor.....1058
Search Scheduler Job.....1058
Searching for a Submitted Scheduler Job.....1059
Viewing a Scheduler Job Details.....1060

Copyright © 2024, Oracle and/or its affiliates.

Cancelling a Scheduler Job.... 1060
Restarting a Scheduler Job.... 1061
Scheduler Job Monitor Dashboard.... 1061
In Progress Scheduler Jobs.... 1061
Failed Scheduler Jobs.... 1062
Completed Scheduler Jobs.... 1062
Upcoming Scheduler Jobs.... 1063
Scheduler Job Details.... 1064
In Progress Scheduler Jobs Details.... 1065
Failed Scheduler Jobs Details.... 1065
Completed Scheduler Jobs Details.... 1066
Scheduler Job Details Zone.... 1066
Viewing a Scheduler Job Details.... 1067
Scheduler Program.... 1067
Search Scheduler Program.... 1068
Searching for a Scheduler Program.... 1069
Defining a Scheduler Program.... 1069
Viewing a Scheduler Program Details.... 1072
Editing a Scheduler Program.... 1073
Deleting a Scheduler Program.... 1075
Scheduler Schedule.... 1075
Search Scheduler Schedule.... 1076
Searching for a Scheduler Schedule.... 1077
Defining a Scheduler Schedule.... 1077
Viewing a Scheduler Schedule Details.... 1081
Editing a Scheduler Schedule.... 1081
Deleting a Scheduler Schedule.... 1084

Appendix A: Administration.... 1087

Feature Configurations.... 1088
Setting the C1_CMO Feature Configuration.... 1088
Setting the C1_FINTRANOP Feature Configuration.... 1090
Setting the C1_PERACCT Feature Configuration.... 1095
Setting the C1-RECLMT Feature Configuration.... 1096

Chapter 1

Person Information

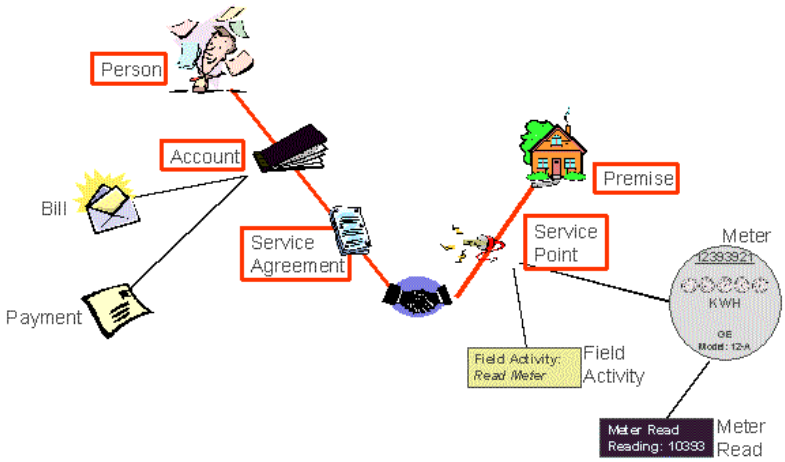
Topics:

- Understanding The V
- Maintaining The V
- Dashboard Portal
- Setting Up Bill Print Groups
- Address
- Start / Stop
- The Big Picture Of Customer Contacts
- Printing Letters
- Maintaining Declarations

We use the term ***person information*** to reference the demographic, geographic, and financial objects that form the core of your CIS system. In this section, we describe how to maintain these objects.

Understanding The V

The "V" is the shape of a diagram we use to illustrate the objects that form the core of the system: **Person**, **Account**, and **Contract**. These objects hold demographic, geographic, and financial information about your company's customers and properties. The following diagram illustrates the objects in the "V", as well as important objects related to the "V" objects (**bill**, **payment**).



You must intuitively understand the concepts embodied in the "V" before you can perform the business processes in the system. In this section, we provide an overview of these objects. In later sections, we describe how you maintain this information.

Persons

A person exists for every individual or business with which your company has contact. Besides customers, persons exist for landlords, contractors, accountants at corporate customers, third party guarantors, collection agencies, etc.

On a person is maintained demographic information like Name, Mailing Address, Phone Numbers, Email Address, Life Support Details, etc.

Most persons are linked to at least one Account because, without an account, the person cannot receive a bill. There are rare examples of persons without accounts, for example, collection agencies that don't receive bills, parent companies that receive statements (as opposed to bills), etc.

Fastpath: Refer to Maintaining Persons on page 4 for more information about persons.

Accounts

Accounts are the entities for which bills are created. You must create at least one account for every customer. The account contains information that controls when bills are created and how they are formatted.

Every account must reference at least one person because the person contains the customer's demographic information (e.g., names, phone numbers, forms of ID). We refer to this individual as the "main" person linked to the account. In addition to the "main" person, an account may reference other types of persons, e.g., the billing contact, the third party guarantor of the account's debt, persons who receive copies of bills, etc.

Most accounts are linked to at least one contract because, without a contract, there is nothing on which a bill can be based. An account without a contract may exist; you just won't be able to do much in the system with such an account.

Fastpath: Refer to Maintaining Accounts on page 12 for more information about accounts.

Copyright © 2024, Oracle and/or its affiliates.

Contracts

Think of a contract as a contract between your company and a customer. The contract contains the terms and conditions controlling how the system calculates charges for the specific service supplied to the customer.

Every account should have at least one contract (otherwise, the account has no financial obligations with your company). There is no limit to the number of contracts that may be linked to an account. However, most financial services customers have X contracts where X is the number of services sold by your company.

A separate contract is required for every service supplied to every account. Why? Because a variety of fields may differ for each contract:

- Obviously, the price of the service is different.
- The credit and collection criteria may be different.
- The general ledger account to which the resultant revenue and/or receivable are linked is different.
- The sales tax could be different for each good / service.
- The list goes on...

Fastpath: Refer to Maintaining Contracts on page 64 for more information.

How To Set Up Customer Hierarchies

Consider the following customer hierarchy:

The above hierarchy shows a parent company (`McDonalds Corporation`) with two subsidiaries and one general manager. Each subsidiary, in turn, has subsidiaries and these subsidiaries can have their own subsidiaries...

You must set up a person for every individual and business in a hierarchy. After the basic demographic information is defined, you define each person's children using the Person to Person page.

Note: A script can be used to set up a hierarchy. The demonstration database contains a script that guides users through the process of setting up a hierarchy.

While hierarchies would typically be set up to define parent companies and subsidiaries, you can use this functionality for many other purposes. For example, a customer relationship manager could set up a hierarchy containing their "important" customers. To do this, they'd need to create a "dummy" person for the parent and then link the "important" customers to it.

Copyright © 2024, Oracle and/or its affiliates.

Maintaining The V

As described in Understanding The "V", the "V" objects are those that form the core of your business processes. In this section, we describe the pages that maintain these objects.

Note: Start / Stop Also Maintains The "V". The Start/Stop Pages also maintain many of the objects in the "V". For example, contracts are created behind-the-scenes when you start service for a customer. This means you won't have to use most of the pages described in this section as part of your day-to-day activities. Rather, they exist in case you need to override what the system automatically set up for you.

Maintaining Persons

On the person page, you define demographic information about your customers and every other individual or business with which your company has contact. The topics in this section describe the person page.

Fastpath: For more information about persons and their place in the "V", refer to Understanding The V on page 2. For more information about how a person is required to set up a customer, refer to Customer Overview.

Person - Main

The Main page contains core person information like names, telephone numbers, and forms of identification. Open this page using **Customer Management, Person, Main**.

Description of Page

Person Information contains basic information about the person. These values only appear after the person exists in the database. The **ID** is a system-assigned random number that stays with a person for the life of the system.

Note: Formatting is performed by a plug-in. The format of **Person Information** is controlled by a plug-in algorithm on the installation record. Refer to the base package's person information algorithm for an example. If you prefer different formatting logic, your system administrator should configure the system appropriately.

The following information may be recorded about a person:

Person Type is used to indicate the type of person (i.e. whether the person is an individual, broker, bill group, and so on.) The valid values are - **Bill Group**, **Broker**, **Business**, **Group**, **Parent Customer**, and **Person**. This field is mandatory. You can define a new person type through the **PER_OR_BUS_FLG** lookup field.

VIP Customer is used to indicate whether the person is a VIP customer. The valid values are - **Yes** and **No**. This field is optional. If you select the **Yes** option from the **VIP Customer** list, the text box corresponding to the **VIP Customer** field is enabled. It allows you to enter the reason why the person is considered as a VIP customer.

Customer Segment is used to indicate the customer segmentation to which the person belongs. This field is optional. You can define a customer segment through the **CUST_SEG_FLG** lookup field.

Customer Tier is used to indicate the tier to which the person belongs. This field is optional. You can define a customer tier through the **CUST_TIER_FLG** lookup field.

Immediate Refund is used to indicate whether you want to process immediate refund for the person's accounts.

Division is used to indicate the division to which the person belongs. This field is optional.

Access Group is used to specify the users who have right to view and update the person's information. The system defaults this value from the user's default access group.

Date of Birth is used to specify the date when the person was born. This field is optional.

Person Since Date is used to specify the date when the person is created in the system. This field is optional.

Copyright © 2024, Oracle and/or its affiliates.

Relationship Manager is used to indicate the user who primarily deals with the person and offers various services to the person. This field is optional. The **Search** (🔍) icon appears corresponding to the **Relationship Manager** field. On clicking the **Search** icon, the **User Search** window appears.

Person Names are used to look for accounts and persons. In addition, a person's primary name is the addressee on the person's bill unless overridden by the Override Mailing Name (maintained on the **Misc** tab). The following fields display:

- Use **Name Type** to indicate if the name is an `Alias`, `Alternate Representation`, `Doing Business As`, `Legal`, or `Primary` name. Note, for new persons, a value of `Primary` is defaulted.

Note: The values for the name type field are customizable using the Lookup table. This field name is NAME_TYPE_FLG.

- Use **Person Name** to define the person's name. Note well, the name is case sensitive.

Note: Alternate representations of a person's name. You would use an `Alternate Representation` for a person's name when you have an alternate ways to define the person's primary name. Alternate representations are typically used in countries that use multiple character sets (e.g., the `Primary` name is entered in Chinese, the `Alternate Representation` is entered in English). When a person has an alternate name, both the main and alternate names can be used to search for a person. The `Alternate Representation` **Name Type** only appears if you have enabled alternate names on the installation record. Refer to the description of the Alternate Representation field under Installation Options - Main for more information.

Note: Validation is performed by a plug-in. The validation that is applied to **Person Name** (e.g., a comma separating the last and first name - `Smith,Patricia`) is controlled by a plug-in algorithm on the installation record. Refer to the base package's name validation algorithm for an example. If you prefer different validation logic, your system administrator should configure the system appropriately.

Person Phone numbers are used to look for accounts and persons. The following fields display:

- **Phone Type** indicates the type of phone number, e.g., Home, Mobile, Business,...
- Use **Phone Number** to define the telephone number. Enter the telephone number in the format described by the **Phone Format**.

Note: Formatting is performed by a plug-in. The format that is applied to a **Phone Number** is controlled by the algorithm that is plugged in on the respective Phone Type. If you prefer a different format, your system administrator should configure this algorithm appropriately.

- Enter the **Extension**, if any, of the telephone number.

A **Person's IDs** have several uses:

- They are used when you look for a customer.
- They are used to highlight potential duplicate persons.
- It displays a person's **primary** identification in the search results area to help a user identify the customer when multiple customers match the search criteria.

Note: Person ID may be required or optional. The person ID usage flag on the installation record indicates whether at least one id for a person is `required` or `optional`.

The following fields are used to define a person's ID(s).

- Turn on **Primary ID** for the piece of **ID** that is the primary means of identifying the customer.
- Indicate the **ID Type**. The **ID Type** defaults from the Installation Record based on the **Person Type** (`Person` versus `Business`).
- Enter the identification number in the adjacent fields. Please note that if the **ID Number** should be formatted (e.g., dashes in an American social security number), you do not have to enter the dashes. Rather, you can enter the information as a contiguous value and the system will format this for you. The format is shown in the adjacent **Identifier Format** column.

Copyright © 2024, Oracle and/or its affiliates.

> **Note: Formatting is performed by a plug-in.** The format that is applied to an **ID Number** (e.g., dashes in an American social security number) is controlled by the algorithm that is plugged in on the respective ID Type. If you prefer a different format, your system administrator should configure this algorithm appropriately.

If you want to add an account for the person when you save the person information, click the **Add Account and Start Service** option. On clicking the **Add Account and Start Service** option, you can specify the customer class in the respective field. The system, by default, sets the customer class depending on the person type. If the person type is set to **Person**, the system defaults the value specified in the **Customer Class (Person)** field of the installation options. However, if the person type is set to **Business**, the system defaults the value specified in the **Customer Class (Business)** field of the installation options. On creating the account, the system links the person to the account using the relationship type which is specified in the **Account Relationship Type Code** field of the installation options. In addition, the Start / Stop screen appears where you can start service for the newly added account.

> **Note:** The **Add Account and Start Service** check box and the **Customer Class** field appear in the **Main** tab of the **Person** screen when the **Create Person Account Together** option type of the **C1_PERACCT** feature configuration is set to **Y**. For more information about the **C1_PERACCT** feature configuration, see Setting the C1_PERACCT Feature Configuration on page 1095.

A message appears in the lower right corner of the page that describes the number of accounts the person has.

Points to Note:

There are certain division-specific validations that are triggered on the **Person** screen. These validations are triggered only when the **Enforce division-specific validations** option type of the **C1_DIVFUNCT** feature configuration is set to **Y**.

The following division-specific validations are triggered on the **Person** screen:

- A person with a division is called a local person. Whereas a person without a division is called a global person. A global person cannot be the main customer of any account in the system.
- You can only assign a division with the status as Active to a person and not the divisions with the status as Pending or Retired.
- The ID Type in the Main tab, the Language in the Contact Information tab and the Characteristic Type in the Characteristics tab that you select must be valid for the division assigned to the person. For example, if you are selecting Division A and English as the language, then the system will check whether the language English is associated with the Division A or not. If the language English is not associated with the Division A while defining the Division A, then the system will throw an error message indicating that the selected language is not valid for the selected division.
- Also, the access group that you select should be valid for the division assigned to the person.
- At anytime, a global person can be converted to a local person provided the selected ID type, Language and Characteristic type are valid for the selected division.
- When you want to change the division assigned to a customer, all products and price lists currently associated to the customer must be valid for the new division.
- A local person can be converted to a global person provided that person is not the main customer of any account in the system.
- For a global person, all products and price lists are valid. There is no division dependency.
- If a local person is defined as the main customer of any account, then the division of that local person cannot be changed.

The following is one more division-specific validation that is triggered on the **Person** screen when the **Create Person Account together** option type of the **C1_PERACCT** feature configuration is set to **Y**:

- The **Add Account and Start Service** check box is enabled when a division is assigned to a person.

Person - Contact Information

This page contains information that may be used to address bills, letters, quotes and statements. Use **Customer Management, Person, Contact Info** to open this page.

Copyright © 2024, Oracle and/or its affiliates.

Description of Page

If the person does not want their primary name (defined on the Main page) used on bills, letters, quotes and / or statements, specify the desired name in **Override Mailing Name 1, 2, and 3**.

Specify the **Override Mailing Address** fields if the person wants their bills, letters, quotes, and / or statements sent to an address other than their service addresses (for example, a post office box). In addition, if the person's account doesn't already indicate that the person's mailing address should be used, you must update this person's account(s). This information resides in the Address Source field on Account - Person.

Note: The **Override Mailing Address** section does not appear when the **Enable Effective Dated Address Feature** option type of the **C1-Address** feature configuration is set to **Y**.

Note: Both the Start / Stop and Order transactions allow you to update the person's **Override Mailing Address** and their account's address source using a single transaction.

Note: Address isn't everything. In addition to defining the person's **Override Mailing Address**, there may be additional tasks you must perform in order to route information to this address. Refer to the following links for more information: where are bills sent, where are letters sent, where are quotes sent and where are statements sent.

If you enter an **Override Mailing Address**:

- The **Country** defaults from your installation options.
- The address constituents may differ depending on the **Country**. Refer to Defining Countries for more information on the address constituents.
- If you have set up postal defaults, the system will default the address constituents when you tab out of the postal code.

Specify the customer's **Email Address** if you communicate with the customer via Email. In addition to defining the person's **Email Address**, you're correspondence routing software must support sending the information via Email. Refer to the following links for more information: where are bills sent, where are letters sent, where are quotes sent and where are statements sent.

Define the **Language** in which the person prefers their bills and correspondence printed.

Note: The person's language defaults from Installation Options - Person.

Fastpath: Refer to Customer Language for more information on options for supporting multiple languages for your customers.

The **Seasonal Addresses** scroll is only used if the person wants their correspondence sent to an alternate address during predefined periods. For example, the customer may want their bills sent to their vacation cottage during the summer. Please be aware of the following:

- Seasonal addresses will only be used if the customer's bills, statements, quotes, and / or letters are routed via the postal service. For example, if you route bills to the customer via Email, the seasonal address will never be used to route bills to the customer. Refer to the following links for more information: where are bills sent, where are letters sent, where are quotes sent and where are statements sent.
- You don't have to specify a seasonal address for every part of the year. For example, if the customer wants their bills sent to their service address except during the summer, you need only enter a seasonal address for the summer.
- You can enter a seasonal address with or without an **Override Mailing Address**. If an **Override Mailing Address** is not specified, the person's correspondence will be addressed as per the instructions on the person's account(s). These instructions reside in the Address Source field on Account - Person.
- You can enter multiple seasonal addresses if the customer so desires.
- The **Season** al periodis defined in the two adjacent fields. The first field contains the day and month when the season starts; the second field contains the day and month when the season ends. The day and month should be entered in the format defined in your display profile.
- The **Status** of the address must be `Active`. You can set the status to `Inactive` if you want a seasonal address ignored (alternately, you can just remove the seasonal address).

Copyright © 2024, Oracle and/or its affiliates.

- The **Country** defaults from your installation options. The address constituents may differ depending on the **Country**. Refer to Defining Countries for more information on the address constituents.
- If you have set up postal defaults, the system will default the address constituents when you tab out of the postal code.

Note: The **Seasonal Addresses** section does not appear when the **Enable Effective Dated Address Feature** option type of the **C1-Address** feature configuration is set to **Y**.

Person - Characteristics

The characteristics page contains information that describes miscellaneous information about the person. Use **Customer Management, Person, Characteristics** to open this page.

Description of Page

Note: You can only choose characteristic types defined as permissible on the person record. Refer to Setting Up Characteristic Types & Their Values for more information.

The following fields display:

Effective Date Define the date on which the characteristic becomes effective. The effective date defaults from the Installation Record.

Characteristic Type Indicate the type of characteristic.

Characteristic Value Indicate the value of the characteristic.

Person - Persons

This page is used to define this person's relationship with other persons. For example, if the person being maintained is a conglomerate, you can define its subsidiaries on this page. Refer to How To Set Up Customer Hierarchies for more information about hierarchies.

Note: You're defining "child relationships". It's important to understand that the persons being defined on this page's grid should be thought of as "children" of the person on the top of the page.

Note: Children may have children. It's possible for one of the children to have children itself (for example, if you have a situation where a subsidiary of a conglomerate itself has subsidiaries). To define a child's children, simply display the child person on this page and then define its children.

Note: Equal relationships. It's possible to link persons where no hierarchy is implied by the relationship (e.g., spouses). There are two ways to do this: 1) you can nominate one spouse as the "parent" and the other as the "child" or 2) you can define the spousal relationship for both persons (i.e., you would define Robert Chopin as the husband of Jeanette Chopin, and Jeanette Chopin as the wife of Robert Chopin). If you choose the latter approach, a recursive relationship will exist.

CAUTION: If your organization enters multiple levels of person, we want to point out that we do not prevent recursive relationships. This means that you could set up a nonsensical situation where person 2 is a child of person 1 and person 1 is a child of person 2.

Use **Customer Management, Person, Persons** to open this page.

Description of Page

The grid contains the "children" associated with the "parent" person who is defined at the top of the page. The following fields display:

Child Person This is the unique identifier of the "child" person. This person's main name appears adjacent.

Child Information An informational message appears to highlight if the child person itself has children. You can click the corresponding go to button to view the child's children.

Relationship Type Indicate the type of relationship between the parent and the child person.

Copyright © 2024, Oracle and/or its affiliates.

Start Date Indicate the date on which this relationship began. This field defaults from the Installation Record.

End Date If the relationship expires, indicate the date the relationship stops.

Person - Web Self Service

This page is used to define information for this person to access account information via your web self-service application.

> **Note: Configuring Web Self Service.** The system provides sample web self-service functionality. This functionality may be used as a basis for implementing a full web application for your organization that is integrated with other web services you offer.

Use **Customer Management, Person, Web Self Service** to open this page.

Description of Page

The information on this page is entered by the customer when registering for the web self service application.

> **Note: Web User ID**. The sample web self service functionality provided by the system defines the customer's web user ID using an entry in the person ID collection defined on the main tab. The **Web Self Service Password** chosen by this person through the web application is stored encrypted and cannot be viewed. Customers should change their password via the web application.

The **Web Self Service Password Hint** and **Web Self Service Password Answer** are defined by the customer when registering and are used when the customer has forgotten the password.

> **Note:** The values for the password hint field are customizable using the Lookup table. This field name is WEB_PWD_HINT_FLG.

The **Web Self Service Receive Marketing Info Flag** indicates whether or not the customer has chosen to receive marketing information. The possible values are `Receives Marketing Info` and `Doesn't Receive Marketing Info`. This information may be used by your web application to optionally send marketing information to this customer via email.

Person - Person To Person Characteristics

The **Person To Person Characteristics** tab allows you to define a characteristic for a person to person relationship. It contains the following fields:

Field Name	Field Description
Person To Person Characteristics	Indicates the child person.
Relationship Type	Indicates how the child person is related to the parent person.
Start Date	Displays the date from when the child person is related to the parent person.
End Date	Displays the date till when the child person is related to the parent person.
Financial Relationship	Indicates whether the child person is financially responsible for the parent person.

> **Note:** The **Next** (⇨) and **Previous** (⇦) buttons allow you to navigate between person to person relationships and accordingly define characteristics for the required person to person relationship.

In addition, this tab contains the following fields in the columnar format:

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
Effective Date	Used to specify the date from when the characteristic is effective for the person to person relationship.	Yes (Conditional) **Note:** This field is required when you are defining a characteristic for the person to person relationship.
Characteristic Type	Used to indicate the characteristic type. **Note:** The list includes only those characteristic types where the characteristic entity is set to **Person to Person**.	Yes (Conditional) **Note:** This field is required when you are defining a characteristic for the person to person relationship.
Characteristic Value	Used to specify the value for the characteristic type. **Note:** If you select a predefined characteristic type, the **Search** (🔍) icon appears corresponding to the **Characteristic Value** field. On clicking the **Search** icon, you can search for a predefined characteristic value. On specifying the value for a predefined characteristic type, the description of the characteristic value appears corresponding to the **Characteristic Value** field.	Yes (Conditional) **Note:** This field is required when you are defining a characteristic for the person to person relationship.

Note: If you want to define more than one characteristic for the person to person relationship, click the **Add** (+) *i*con and then specify the details. However, if you want to remove a characteristic from the person to person relationship, click the **Delete** (🗑) icon corresponding to the characteristic.

Person - Usage Amount/Counter

The **Usage Amount/Counter** zone allows you to search various amount and counter related information, such as average daily balance, average monthly balance, number of auto transfer transactions, and so on, of the person on a particular date and time. It contains the following two sections:

- **Search Criteria** - The **Search Criteria** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Effective From Date	Used to search amount or counter details of the person from a particular date onwards.	No
Effective To Date	Used to search amount or counter details of the person till a particular date.	No
Usage Amount	Used to search a particular type of amount details. The valid values are: • Combined Average Daily Balance • Combined Average Monthly Balance • Combined Balance • Combined Monthly Balance	No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: You can add the required amount types in the **BAL_FLG** lookup field.	
Usage Counter	Used to search a particular type of counter details. The valid values are: • No. of ATM transactions • No. of Auto Transfer	No
	Note: You can add the required counter types in the **CNT_FLG** lookup field.	
Usage Type	Used to indicate whether you want to search amount or counter details. The valid values are: • Usage Amount • Usage Counter	No

Note: If you search the amount or counter details without specifying any criteria, the system lists the amount and counter details of the person which are available on the current date.

- **Search Results** - On clicking the **Search** button, the search results appear based on the specified search criteria. The **Search Results** section contains the following columns:

Column Name	Column Description
Effective Date	Displays the date for which the amount or counter details are available.
Usage Type	Indicates whether the amount or counter details are available. The valid values are: • Usage Amount • Usage Counter
Usage Criteria	Indicates the type of amount or counter details.
Usage	Displays the amount or counter depending on the usage criteria.
Currency	Indicates the currency in which the amount is available.
	Note: The data does not appear in this column when the counter details are displayed.

Person - Deal Information

The **Deal Information** tab allows you to view the deal information of the person received from the external system. It contains the **Deal Information** zone which has the following columns:

Column Name	Column Description
Deal ID	Displays the deal ID. This is an identifier generated in the Deal Pricing Management system (such as, Oracle Revenue Management and Billing Analytics (ORMBA)).
Currency	Indicates the currency in which the deal is made.
Prospect	Indicates whether the person is a prospective customer or not. The valid values are **Yes** or **No**.

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
Multi-Division	Indicates whether the deal is applicable to multiple divisions. The valid values are **Yes** or **No**.
Multi-Deal	Indicates whether the deal is a part of a multi-buy deal. The valid values are **Yes** or **No**.
Start Date	Displays the date from when the deal is effective.
End Date	Displays the date till when the deal is effective.
Deal Review Frequency	Indicates the interval at which the deal is reviewed. The valid values are **Annual**, **Monthly**, and **Weekly**.
Relationship Manager	Indicates the professional who is working to improve the relationship with the person with respect to the deal.

Person - Address

The **Address** tab allows you to view a list of addresses associated with the person. It appears when the **Enable Effective Dated Address Feature** option type of the **C1-Address** feature configuration is set to **Y**. It contains the following zones:

- Search Address on page 111
- Address Characteristics on page 116
- Address Entities on page 115

You can also define an address for the person by clicking the **Add** button in the **Page Title** area of the **Person** screen.

Note: When you are defining an address for the person from the **Person** screen, the system, by default, specifies the person details in the **Address Entities** section.

Related Topics

For more information on...	See...
How to setup the **C1-Address** feature configuration	Setting the C1-ADDRESS Feature Configuration

Maintaining Accounts

The account record contains information that controls billing and collection processes. You should only need to use this page if you need to fine-tune the information that the system has set up by default because:

- Most accounts are created when you add a new person or when you complete an order for a new customer.
- Many fields are populated behind-the-scenes as part of the service activation process.

The topics in this section describe the pages on which account related information is maintained.

Fastpath: For more information about accounts and their place in the "V", refer to Understanding The V on page 2.

Account - Main

The Main page contains core account information. Open this page using **Customer Management, Account**.

Description of Page

The primary name of the **Account**'s main customer and the **Account ID** are displayed on every tab in this page. These values only appear after the account exists on the database. The **Account ID** is a system-assigned random number that stays with an account for life.

Copyright © 2024, Oracle and/or its affiliates.

Note: Formatting may be performed by a plug-in. The basic information about an account that appears at the top of this page (and on many other pages in the system) may be formatted by a plug-in algorithm on the installations record. Refer to the base package's account information algorithm for an example. If you prefer different formatting logic, your system administrator should configure the system appropriately.

A "check digit" is displayed adjacent to the account ID. This is for information purposes only, and is not needed to operate the system. The following points describe how the check digit is calculated for an Account ID equal to `0011883422`.

- Calculate the sum of the first and every alternate digit in the account id. A = 14 = 0+1+8+3+2
- Calculate the sum of the second and every alternate digit in the Account ID and multiply by 2. For example, B = 30 = (0 + 1 + 8 + 4 + 2) * 2
- Add A and B. For example, C = 44 = 14 + 30
- Count the number of digits used to calculate B that are greater than 4. For example, D = 1 since only 8 is greater than 4.
- Multiply D by 9. For example, E = 9 = 1 x 9
- Subtract E from C. For example, F = 35 = 44 - 9
- Subtract the units position of F from 10 to find the check digit. For example, check digit = 5 = 10 - 5 (5 is subtracted from 10 because F = 35 and there the units position is 5).

Note: Technical note. The above is calculated in the common routine called `CIPCACDN`.

Customer Class plays a part in:

- If and when a customer is subject to late payment charges.
- The account's default collection class and when the account debt monitor reviews an account. Refer to The Big Picture of Credit & Collections for more information about how and when an account's debt is reviewed.
- And several other functions. Refer to and Setting Up Customer Classes for more information.

Note: The customer class defaults from Installation Options - Account (Person Customer Class) and may be overridden at will.

Division defines the jurisdiction that governs this account. You may only select **Divisions** associated with the account's *Customer Class*. This field is updated behind the scenes every time a contract is activated (the system uses the **Division** associated with the contract's contract type). If you have assigned a **Division** and do not want the system to change it when a contract is activated, turn on **Protect Division**.

Note: Division governs many functions. An account's division impacts its subsequent bill due dates, late payment charge dates, credit & collections review dates, its default budget plan, the roles assigned to To Do entries, and the calendar of workdays. Refer to Setting Up Divisions and Setting Up Customer Classes for more information.

Set Up Date is the date the account was initially set up. This is purely informational.

Invoice Currency defines the currency in which the account's financial transactions are expressed. All rates and payments associated with this account must be denominated in this currency. You can change the invoice currency of an account whenever required to meet the changing business needs. For example, when a customer opens an account, it might be billed in US Dollars. However, with the Multi-Currency Accounts feature enabled, the same account can be billed in Euros in the future.

You can change the invoice currency of an account when:

- You have set the **C1_MLTCURACC** feature configuration.
- The account's customer class is enabled for open item accounting. In other words, the **Open Item Accounting** check box of the account's customer class is selected.
- The account does not have any active or pending loans, payment plans, payment arrangements, or recurring charge contracts.

For more information about the Multi-Currency Accounts feature, refer to Multi-Currency Accounts.

Default note. The currency defaults from the Installation Record and may be overridden here.

Copyright © 2024, Oracle and/or its affiliates.

Access Group controls which users are allowed to view and update this account's information (including bills, contracts, and payments). The system defaults this value from the user's *default access group*. Refer to The Big Picture of Row Security for a complete description of how account security is implemented in the system.

The optional Account **Management Group** controls the roles assigned to To Do entries associated with an account. Refer to Setting Up Account Management Groups for more information.

Account Category allows you to indicate the category to which the account belongs. The valid values are:

- **Invoice Account** - Indicates that the account will be used to bill charges of the usage accounts. It will not have charges of its own.
- **Settlement Account** - Indicates that the account will be used only for settlement purposes.
- **Usage Account** - Indicates that the account will have its own charges. If the usage account is linked to any construct, the usage account's charges are billed through an invoice account. Otherwise, the usage account is billed individually.

If you do not specify the account category while creating an account, by default, the account category is set to **Usage Account**.

Immediate Refund indicates whether you want to process immediate refund for the account.

Eligible for Repricing indicates whether the account is eligible for repricing when the account attributes (defined as parameters in the system) are changed.

Note: On creating a new account where the **Eligible for Repricing** option is selected, an entry is not created in the **CI_REPRC_REQ_DTL** table. In other words, the repricing is not triggered on adding a new account. Therefore, you need to manually execute the **C1-PriceAccount** business service to calculate and persist the rate for the account.

Product is used to specify the product for which the account is created for the main customer. For example, a bank may offer wide range of loan products (such as Car Loan, Personal Loan, House Loan, and so on) and a person has opened the car loan account with the bank. In such case, the Car Loan product is mapped to the account. The **Search** (🔍) icon appears corresponding to the **Product** field. On clicking the **Search** icon, the **Product Search** window appears.

Note:

On searching a product in the **Product Search** window, the search results include only those products which are associated with the division to which the account belongs.

When you create an account and map it to a product, the division-specific default price list defined for the product is assigned to the account. The price item pricing on the default price list will be available to the new account when you manually execute the **C1-PriceAccount** business service.

By default, the account is eligible for accrual when the division to which the account belongs is eligible for accrual. If you do not want the account to be eligible for accrual, you must select the **Exclude Accrual** check box. On selecting the **Exclude Accrual** check box, the **Accrual Cycle** field is disabled.

By default, the accrual cycle defined for the division appears in the **Accrual Cycle** field when the division to which the account belongs is eligible for accrual. You can override the accrual cycle, if required.

Use the **Closing Date** field to specify the date from when you want to close the activities on the account. If the account closure feature is enabled, the closing date is recalculated considering the grace days, if any. The account is not billed from the recalculated closing date onwards. For more information about the feature, see Account Closure on page 27.

Enter a **Comment** to define unusual information about the account.

Bill Cycle controls when a bill is produced for an account. If a characteristic type which is specified in the **Bill Cycle Rule Code** option type of the **C1-ASOBLLNG** feature configuration is defined for an account, the system derives the bill cycle for the account using the billing preference. For more information, see Billing Preference.

If you have assigned a bill cycle and do not want the system to change the bill cycle when a contract is activated, turn on **Protect Bill Cycle**.

Copyright © 2024, Oracle and/or its affiliates.

Bill On or After Date – Used to specify the date after which bills can be generated for the account. The system will hold the bill generation process for the account till the specified duration. It will allow to create bills for the account from the user interface only after the specified date or through a batch process only on or after the specified date.

If user should review the account's printed bills before they are sent to the customer, enter the user ID of the individual who reviews the bill in **Bill Print Intercept**.

The balances shown in the scroll are the total amounts of debt belonging to every debt class linked to the account's contracts. The following information is displayed:

- **Debt Class** is the debt class associated with the account's contract(s).
- **Current Balance** is the amount of debt the customer currently owes.
- **Payoff Balance** is the amount the customer would owe if they wanted to pay off their debt.

To see summary information about the contracts that contribute to the financial totals click the go to button adjacent to the respective amount.

CAUTION: If you do not understand the difference between payoff balance and current balance, refer to Current Amount versus Payoff Amount.

The **Trial Bill** check box allows you to indicate whether the account should be considered during trial billing when the C1-PNDBL, C1-BSGEN, and C1-BLPPR batches are executed with the **Process All or Selected Accounts** parameter set to **Y**.

Bill Lead Days - Used to indicate whether you want to bill the account in advance. You need to specify the number of days which is used to derive the cut-off date for advance billing. In case you don't want to bill the account in advance, you must specify **0** in this field.

Points to Note:

There are certain division-specific validations that are triggered on the **Account** screen. These validations are triggered only when the **Enforce division-specific validations** option type of the **C1_DIVFUNCT** feature configuration is set to **Y**.

The following division-specific validations are triggered on the **Account** screen:

- The division to which the account belongs should be the same as the division to which the person belongs if that person is the main customer of that account.
- The Invoice Currency and Bill Cycle in the Main tab, the Bill Message in the Bill Messages tab, the Characteristic Type in the Characteristics tab, and the Account Identifier Type in the Account Identifiers tab must be valid for the division assigned to the account. For example, if you are selecting Division A and the invoice currency as USD, then the system will check whether USD is associated with Division A or not. If the invoice currency USD is not associated with Division A, then the system will throw an error message indicating that the selected invoice currency is not valid for the selected division.
- Also, the access group that you select should be valid for the division assigned to the account.
- In an account, a local person from different division can be a miscellaneous customer of an account.
- When products and price lists are associated with an account, they must be valid for the division assigned to the account.
- When you want to change the division assigned to an account, all products and price lists currently associated to the account must be valid for the new division.

Account - Auto Pay

The Auto Pay page is used when an account pays their bills automatically (e.g., by direct debit or credit card).

Fastpath: Refer to How And When Are Automatic Payments Created? on page 306 for more information.

Open **Customer Management, Account** and navigate to the **Auto Pay** tab to maintain this information.

Description of Page

Copyright © 2024, Oracle and/or its affiliates.

Defer Auto Pay Date - Used to specify the date after which automatic payments can be created for the account. The system will hold the automatic payment process for the account till the specified duration. It will allow to create automatic payments for the debit bills through a batch process only after the specified date.

Use the **Rule Based Auto Pay** option when you want to define a default, manual, or rule based auto pay instruction for the account.

Note: You can select the **Rule Based Auto Pay** option when all existing auto pay instructions created from the **Account** screen are past dated. And, you can deselect the **Rule Based Auto Pay** option when all existing auto pay instructions created from the **Auto Pay Instruction** screen are past dated.

The **Account Auto Pay** scroll defines the bank account / credit card from which the customer's automatic payments are debited. Multiple auto-pay options may exist if the customer changes auto-pay options over time.

Note: Installation Option Algorithm for autopay amount calculation should be set to APAM-SETT (CIPPKDSX).

The **Auto Pay Usage** field is used to indicate whether the automatic payment option is available for automatic credit or debit, or for both automatic credit and debit.

The following fields display:

- **Start Date** is the date on which the automatic payment information comes into affect. The system creates an automatic payment for any bill produced for this account with a due date on/after this date and on/before the **End Date**. If **End Date** is not specified, this means the automatic payment option applies indefinitely. You need only specify an **End Date** if the customer wants to stop paying automatically.
- **Auto Pay ID** is the unique identifier of the auto-pay record.
- **Percentage** is the contribution of payment amount in percentage within multiple Auto Pay IDs (i.e for Auto Pay Sources) having the same Priority.
- **Priority** is a number representing priority for a set of auto pay sources and a auto pay route type for a given start and end date with other set of Auto Pay Sources and Auto Pay Route Types.
- Use **Auto Pay Source Code** to define the source of the funds used to satisfy the automatic payment request. For example, if a customer indicates that they want to use their checking account to pay their bill, you would specify the **Auto Pay Source Code** associated with their bank. The source's description and external source ID (e.g., bank routing number) are displayed adjacent.
- Use **Auto Pay Method** to specify whether you want the system to process automatic payments as `Direct Debit` or `Payment Advice`. This field is visible only if feature configuration is set up to support payment advice functionality. Refer to Payment Advices for more information on the payment advice functionality.
- Use **Auto Pay Route Type** to control when and how automatic payment requests of the account are routed to a financial institution. It also controls when the general ledger is impacted by the automatic payment financial transaction.

  Fastpath: Refer to Setting Up Auto-Pay Route Types for more information.
- Use **Account Number** to define the customer's bank account / credit card number.
- You will be required to define an **Expires On** date if the **Auto Pay Source Code** references a tender type that requires an expiration date (e.g., if the Auto Pay Source is a credit card company).
- Enter the **Name** of the customer as it appears in the financial institution's system. This name is routed to the financial institution.

Note: The **Name** will default to the primary name of the main customer linked to the account after you enter a **Start Date**. This defaulting is only possible for accounts that exist on the database.

- In some locales, customers can define a **Maximum Withdrawal Amount** to limit the amount of money that is automatically debited from their account. Refer to How To Implement Maximum Withdrawal Limits on page 308 for more information.
- Use **Comments** to describe anything interesting / unusual about the automatic payment request.

Copyright © 2024, Oracle and/or its affiliates.

Fastpath: Refer to How To Split Automatic Payment on page 310 for more information on how multiple auto pay sources are allowed for given start date and end date of an Auto Pay record. For payment amount calculation in split auto pay, refer to Installation Options - Algorithms where CI_APAMSETT has to be configured.

Account - Persons

The Account Person page contains information about the person(s) linked to the account.

Note: Both customers and non-customers can be linked to an account.

When you add an account from Person - Main or from Order - Main the system automatically links the person to the account, so you typically won't have to open this page. However, if you need to link multiple persons to an account (e.g., because you need to send a duplicate of the account's bill to someone other than the main customer) or change information about a person on an account, open **Customer Management, Account** and navigate to the **Persons** tab.

Description of Page

The **Account Persons** scroll contains one entry for every person linked to the account. The following fields display:

Person ID The unique identifier of the person linked to the account. The person's name is displayed adjacent.

Main Customer Turn on this switch if the person is the main customer on the account. Only one person on an account may be designated as the main customer.

The significance of the main customer is that its name will appear adjacent to the account ID throughout the system.

Financially Responsible Turn on this switch if the person is financially responsible for the account's debt.

This switch is on and gray if the person is the **Main Customer** because the main customer is always financially responsible. If multiple persons are linked to the account, you use this switch to indicate which ones are financially responsible versus those that are linked for other purposes.

Third Party Guarantor Turn on this switch if the person is a 3 rd party guarantor of the account's debt. This switch is off and gray if the person is the **Main Customer**.

Relationship Type Define the relationship between the person and the account. Refer to Setting Up Account Relationship Codes for information about setting up relationship types.

Note: When the system links the person to the account when you add an account from Person - Main, it defaults the relationship type from the Installation Record.

Web Access Indicate whether or not a person linked to this account is `Allowed` or `Not Allowed` to view details of this account via your web self service application.

Note: This flag is defaulted to `Allowed`.

Prefix/Suffix and **Pfx/Sfx Name** Use these fields if you need to append additional information to a customer's name when correspondence or bills are sent to this person. For example, if the account has declared bankruptcy you might attach "Debtor In Possession" to the person's name. Use **Prefix/Suffix** to indicate if the **Pfx/Sfx Name** should be appended to the front or the back of the customer's name.

The **Address Information** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Country	Used to indicate the country.	Yes

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: The remaining fields in this section appear depending on which address fields are turned on while defining the country. Also, the labels of the address fields will change depending on the business label defined for the address field in the **Country** screen.	**Note:** This field is required when you have selected the address source as **Account Override** in a bill routing information defined for the person which is linked to the account.
Address 1	Used to specify the first line of the address. It may contain details, such as house number and apartment name.	No
Address 2	Used to specify the second line of the address. It may contain details, such as street name.	No
Address 3	Used to specify the third line of the address. It may contain the landmark details.	No
Address 4	Used to specify the fourth line of the address.	No
House Type	Used to indicate the type of the house.	No
Number 1	Used to specify the numeric information, if any, related to the address.	No
Number 2	Used to specify the numeric information, if any, related to the address.	No
In City Limit	Used to indicate whether the address is within the city limit or not. The valid values are: • Yes • No	No
City	Used to specify the city name.	No
Geographic Code	Used to specify the geographic code of the address.	No
County	Used to specify the county name.	
State	Used to indicate the state where the city or county is located. **Note:** The states listed in this field change depending on the country that you have selected.	No
Postal	Used to specify the postal or zip code of the address.	No

In the **Bill Routing Information** section, you can define multiple bill routing records for the person which is linked to the account. For each bill routing record, you can specify the following:

- **Receives Copy of Bill** - Turn on this switch if the person receives a copy of the account's bills. Turning off this switch grays the remaining fields. This switch is on and gray if the person is the **Main Customer** because the main customer always receives a copy of the bill.

  Refer to The Source Of Bill Routing Information on page 180 for more information about how the remaining information is used to format the address of a bill.
- **Bill Route Type** - Indicate how the bill is sent to the customer. This field's value defaults from the Installation Record. If the **Bill Route Type** you select indicates that bills are routed via Fax, the person's fax number is displayed adjacent (the system knows which of a person's phone numbers is a fax number by the phone type). If the **Bill Route Type** you select indicates that bills are routed via Email, the person's *Email address* is displayed adjacent. If the

Copyright © 2024, Oracle and/or its affiliates.

Bill Route Type you select indicates that bills are routed via the Postal service, you must choose an appropriate **Address Source** to define which address should be used. Refer to Setting Up Bill Route Types for more information about bill route types.

- **Bill Format** - Indicate if the customer should receive a `Detailed` or a `Summary` bill.

> **Note:** The values for this field are customizable using the Lookup table. The values need to match the formats supported by your bill print software. This field name is BILL_FORMAT_FLG.

- **Number of Bill Copies** - Indicate how many copies of the bill the person receives.
- **Customer PO ID** - Indicate the purchase order ID the customer wants printed on their copy of the bill.
- **Receives Copy of Quote** - Turn on this switch if the person receives a copy of the account's quotes. Turning off this switch grays the remaining fields. This switch is on and gray if the person is the **Main Customer** because the main customer always receives a copy of the account's quotes.

> **Fastpath:**
>
> **Receives Copy of Quote** only appear if the quotation `Contract Management` module is not turned off.
>
> Refer to Printing Quotes on page 779 for more information about how the remaining information is used to format the address of a quote.

- **Quote Route Type** - Indicate how the quote is sent to the customer. This field's value defaults from the Installation Record. If the **Quote Route Type** you select indicates that quotes are routed via Fax, the person's fax number is displayed adjacent (the system knows which of a person's phone numbers is a fax number by the phone type). If the **Quote Route Type** you select indicates that quotes are routed via Email, the person's *Email address* is displayed adjacent. If the **Quote Route Type** you select indicates that quotes are routed via the Postal service, you must choose an appropriate **Address Source** to define which address should be used. Refer to *Setting Up Quote Route Types* for more information about quote route types.

> **Note:** **Quote Route Type** only appear if the quotation `Contract Management` module is not turned off.

- **Receives Notification** - Turn on this switch if the person receives letters (i.e., notifications) initiated by collection, write-off, and overdue events. These events will send letters to the main customer and any other person linked to the account that **Receives Notifications**. This switch is on and gray if the person is the **Main Customer** because the main customer always receives notifications. Refer to Printing Letters for more information about how letters are produced by the system.
- **Address Source** - If the **Bill Route Type** or the **Quote Route Type** you selected indicates that bills / quotes are routed via the postal service, you must choose the appropriate **Address Source**.

  Choose `Person` if bills / quotes should be sent to the person's mailing address. This address is displayed adjacent.

  Choose `Account Override` if bills / quotes should be sent to an override address instead of the person's mailing address. Typically, you would only choose this option if the person has multiple accounts and each account's bills / quotes should be sent to a different address. If you have selected the address source as `Account Override`, you must either specify the override address in the **Address Information** section or the address ID in the **Bill Routing Information** section.
- **Address ID** - Used when you want to send the bills or quotes generated for the account to a particular address. If the address source is set to **Person**, you can only specify the address ID which is associated with the person. It can be either the person's mailing or seasonal address. If the address source is set to **Account Override**, you can only specify the address ID which is associated with the account. It can be only the account override address.

Copyright © 2024, Oracle and/or its affiliates.

> **Note:**
>
> The **Search** (🔍) icon appears corresponding to this field. On clicking the **Search** icon, the **Address Search** window appears where you can search for the required address.
>
> If the address ID is not specified and address source is set to **Person**, the mailing address which is effective for the person at the time of billing is considered for bill routing. Similarly, if the address ID is not specified and address source is set to **Account Override**, the account override address which is effective for the account-person combination at the time of billing is considered for bill routing.

Account - Financial Balances

The financial balances page is an information-only page that subdivides an account's debt by debt class. The balances shown on this page are the total amounts of debt belonging to every debt class linked to the account's contracts.

> **Note: The source of debt class.** An account's debt comes from its contract(s). Every contract has an contract type. Every contract type references a debt class.

Open **Customer Management, Account** and navigate to the **Financial Balances**.

Description of Page

The account's **Current Balance** displays at the top of the page.

The **Debt Class Balances** scroll contains one entry for every debt class linked to the accounts non-closed / non-cancelled contracts. The following information is displayed for each debt class.

- **Debt Class** is the debt class associated with the account's contract(s). To see summary information about the contracts that contribute to the debt class totals click the adjacent Go To button.
- **Current Balance** is the amount of debt the customer currently owes.
- **Payoff Balance** is the amount the customer would owe if they wanted to pay off their debt. Payoff balance is only displayed if it differs from current balance.

> **CAUTION:** If you do not understand the difference between payoff balance and current balance, refer to Current Amount versus Payoff Amount.

- The **Debt Class Arrears** shows the age of all amounts that make up the current balance.
- The **Days Old** and **Amount** information shows the age of the debt class's debt.

> **Note:** Information about debt balances by age is not applicable to open-item accounts.

Account - Bill Messages

The Account Message page contains information about the message(s) to appear on an account's bills. Both one-time and permanent messages may be defined. This information is used when the system assembles messages to appear on an account's bill. Refer to The Source Of Bill Messages on page 181 for more information about bill messages.

Open **Customer Management, Account** and navigate to the **Bill Messages** tab to access this page.

Description of Page

The following fields are displayed:

Bill Message The message code and description of the message that appears on the customer's bill.

Bill Message Type Use `Temporary` to indicate the message should only be linked to the next bill produced for the account. Use `Permanent` if the message should appear on every bill. Note, a value of `Temporary` defaults.

Copyright © 2024, Oracle and/or its affiliates.

Account - C&C

The Credit Rating page contains the credit rating transactions that impact an account's credit rating and cash-only score. To view or modify these transactions, open **Customer Management, Account** and navigate to the **C&C** (credit and collections) tab.

Description of Page

Collection Class controls how the account's debt is compared against collection criteria to determine if a collection process should be started. Refer to The Big Picture Of Credit & Collections (C&C) for more information about how and when an account's debt is reviewed.

> **Note:** The **Collection Class** defaults from the account's customer class when the account is first added. It may be overridden at will.

Postpone Credit Review Until Date – Used to specify the date after which the account's debt can be reviewed by the credit and collections processes (i.e., the account debt monitor, the write-off monitor, and the delinquency monitor). The system will hold the overdue or delinquency process for the account till the specified duration. It will allow to initiate an overdue or delinquency process for the account only after the specified date.

Last Credit Review Date is the date when the account's debt was last reviewed by the account debt monitor (ADM).

> **Note: Forced ADM review.** If you want the ADM to review an account's debt the next time it runs, simply change the **Credit Review Date** to a date in the distant past (e.g., 1/1/1900). The next time the ADM2 background process runs, it'll see that a great deal of time has past since the last time it reviewed the account's debt and review it (and then set the **Credit Review Date** accordingly).

The account's **Current Credit Rating** is displayed. For an explanation of how this number is derived, see How is an account's credit rating calculated?.

The account's **Current Cash Only Score** is displayed. For an explanation of how this number is derived, see How is an account's cash-only score calculated?.

The **Credit Rating History** scroll contains one entry for each credit rating history record associated with the account. The following fields display:

Start Date This is the first date the credit rating transaction affects the account's credit rating and cash-only score.

End Date This is the last date the credit rating transaction affects the account's credit rating and cash-only score.

The `Created On` message is formatted using an algorithm on the Installation Options.

Affect Credit Rating By This is the effect of the credit rating transaction on the account's credit score. This should be a *negative* number because the *lower* the score the worse the credit rating.

An account's credit rating is equal to the start credit rating amount defined on the installation record plus the sum of credit rating demerits that are currently in effect.

Affect Cash-Only Score By This is the effect of the credit rating transaction on the account's cash-only score. The *higher* the score, the greater the chance this account will be marked as cash-only. Therefore, a *positive* number should be used to make the score go up.

An account's cash-only score is equal to the start cash-only score defined on the installation record plus the sum of cash-only points that are currently in effect. When an account's cash-only score exceeds the cash-only threshold score defined on the installation record, the account is flagged as cash-only on the payment event page.

Comments Enter comments to clarify the reason(s) for the creation of the transaction.

- How are credit rating transactions created? on page 22
- How is credit rating calculated? on page 22
- What impact does credit rating have in the system? on page 22
- How is cash-only score calculated? on page 22
- What impact does cash-only score have in the system? on page 22

Copyright © 2024, Oracle and/or its affiliates.

How are credit rating transactions created?

Credit rating transactions may be created as follows:

- Certain types of collection events add credit rating transactions. The number of credit rating and cash-only points is defined on the collection event type.
- Certain types of write-off events add credit rating transactions. The number of credit rating and cash-only points is defined on the write-off event type.
- If a payment tender is canceled with a cancellation reason that indicates the cancellation was due to non-sufficient funds, a credit rating transaction is created. The number of credit rating and cash-only points is defined on the payment cancellation reason code.
- You may configure your system to create a credit rating transaction when a promise to pay is broken. To do this, you must plug in an appropriate Broken Algorithm on the promise to pay type.
- A user may create credit rating transactions at their discretion.
- Algorithms may add credit rating transactions. The system provides sample overdue event activation algorithms that add credit rating transactions. Refer to C1-OE-CR-RT and C1-CE-CR-RT for more information. Some examples of algorithms your implementation may decide to introduce: the creation of a late payment charge. To do this you could introduce a new adjustment freeze algorithm.

Note: Created By Flag. The credit rating transaction includes a **Created By** flag that is customizable using the Lookup table. This field name is CR_RAT_CRE_BY_FLG. If your implementation uses algorithms to create credit rating history transactions, you may consider adding appropriate values to the Created By flag. Also note that the Created By flag includes the value `Other`, which may be used by your implementation specific algorithms if you do not want to add a new custom value.

How is credit rating calculated?

An account's credit rating is calculated by summing the Affect on Credit Rating from credit rating transactions effective on the current date. This value is added to the perfect credit rating defined on the installation record. The result is the customer's current credit rating.

What impact does credit rating have in the system?

The following points describe the impact of an account's credit rating:

- An account's current credit rating is displayed in the Control Central Alert area when it's less than the threshold credit rating defined on the installation record.
- An account's credit rating may affect how the account's debt is collected. We used the word "may" because an account's credit rating will only affect collection criteria if you set up your collection processes to do this.

Fastpath: For more information, refer to Designing Your Collection Class Control Overrides.

How is cash-only score calculated?

An account's cash-only score is calculated by summing the Affect on Cash-Only Score from credit rating transactions effective on the current date. This value is added to the beginning cash-only score defined on the installation record. The result is the customer's current cash-only score.

What impact does cash-only score have in the system?

If an account's current cash-only score exceeds the cash-only threshold on the installation record, and the base package's C1-CASH-ACCT algorithm is enabled on the installation record:

- The customer is flagged as cash-only in the Control Central Alert area.
- If a user attempts to add a payment whose tender type is a cash-equivalent, a warning is issued.

Account - Budget

The Budget page contains information about all contracts linked to an account that are eligible to participate in a budget billing plan. Refer to Budget Billing for more information about how this information affects the account's bills.

Copyright © 2024, Oracle and/or its affiliates.

Note: You are actually changing contracts. When you change a contract's budget amount on this page, you are actually changing the contract's recurring charge amount. Refer to Contract - Recurring Charge for more information.

Note: Eligible contracts. It's important to be aware that some contracts linked to an account may not be eligible for budget plans. For example, assume you have a customer with a deposit contract and one or more utility contracts. This customer will be billed for their deposit regardless of the utility contracts' budget amounts. You define whether a contract may participate in a budget plan on the contract's contract type.

Open **Customer Management, Account** and navigate to the **Budget** tab to view this information.

Description of Page

The appearance and dialogue of this page differs slightly for an account with a small number of contracts (100 or less) versus a large number of contracts (more than 100).

- For an account with a small number of contracts, the user may click the **Recommend Budget** button to generate a **New Budget** for each contract. At that point the contract's budget amount has not been updated. The user may review the calculated amounts; make desired changes and then click Save to update the budget amount for the contracts.

CAUTION: You must save the page in order for the budget changes to take affect.

- For an account with a large number of contracts, the user may click the **Calculate and Apply New Budget** button. At that point the system will calculate the recommended budget and update the **Budget Amount** on all the eligible contracts. To change the budget amount for one or more of the contracts, the user navigates to the contract page using the drill down button adjacent to the contract in question.

Click **Cancel Budget** to have the system set the **New Budget** amount to zero for all the account's contracts on budget.

Note: Budget Recommendation. Though this page lists all contracts eligible for a budget, the budget recommendation only calculates a budget amount for contracts whose start and end date (if any) includes the budget date. Additionally, the budget date is used to select the rate schedule that is in effect for an contract if estimating consumption is necessary.

Budget Plan controls how the account's debt is managed for budget-related purposes. Specifically, it controls how the recommended budget amount is calculated and how / when the customer's budget is periodically changed.

Note: Budget plan defaults based on the account's customer class / division combination.

Use **New Budget Date** to define the effective date of any changes to the customer's budget amount (this date is the effective date on the respective contracts' recurring charge history).

The system displays the sum of the contracts' current budget amounts in **Total Budget Amount**.

For accounts with a small number of contracts, the system displays the sum of the contracts' new budget amounts in **Total New Budget**.

The grid that follows contains a row for each contract linked to the account that is eligible to participate in a budget billing plan:

Contract Information This is a concatenation of basic information about the contract.

Last Changed Date This is the date when the contract's budget amount (i.e., recurring charge amount) was last changed.

Budget Amount This is the existing budget amount (i.e., recurring charge amount) for the contract.

New Budget For accounts with a small number of contracts, the contract's budget amount (i.e., recurring charge amount) will be changed to reflect this value when the page is changed. The effective date of the change is defined in **New Budget Date**.

Account - Deposits

The Deposits page contains information about all cash and non-cash deposits linked to an account.

Copyright © 2024, Oracle and/or its affiliates.

> **Note: Cash versus Non-cash Deposits.** Non-cash deposits (e.g., letters of credit, surety bonds, 3 rd party deposits) can be added and maintained on this page. Cash deposits are maintained using contracts.

Open **Customer Management, Account** and navigate to the **Deposits** tab to view this information.

Description of Page

The following fields appear in the **Deposit Summary**:

Cash Deposits On Hand The total amount of cash deposits currently held for the account. This value is equal to the sum of the payoff balances from all contracts whose contract type has a special role of `Cash Deposit`.

Cash Deposits Requested This value is equal to the sum of Total Amount to Bill from all contracts whose contract type has a special role of `Cash Deposit`.

Non-Cash Deposits The total amount of active non-cash deposits currently held for the account. This value is equal to the sum of non-cash deposits displayed in the scroll below (note, non-cash deposits can be maintained in the same scroll).

Total Deposits on Hand This is the sum of cash and non-cash deposits held for the account.

Turn on **Do Not Review or Refund** if this account should not be considered by the Deposit Review background process (this process highlights customers who require an additional deposit) or by the Deposit Refund background process.

The **Non-Cash Deposit** scroll contains an entry for every non-cash deposit linked to the account.

To modify a non-cash deposit, simply move to a field and change its value. To add a new non-cash deposit, click the + button and then fill in the information for each field.

Start Date This is the date the non-cash deposit is effective.

End Date This is the expiration date of the non-cash deposit.

Non Cash Deposit Type Select the appropriate non-cash deposit type. The value selected is important as it controls whether the system requires a **Third Party Deposit** and whether the system will create a To Do entry near the expiration date of the deposit. Refer to Setting Up Non-Cash Deposit Types for more information.

Deposit Class Select the deposit class in which this non-cash deposit should be included. This value is used when the system amalgamates an account's total deposits when checking if the account holds an adequate deposit (amalgamation and recommendation happens within a deposit class).

Deposit Amount Indicate the amount of the non-cash deposit.

Document Number Enter the document number, if any, associated with the non-cash deposit (e.g., the letter of credit number, the surety bond number, etc.).

Third Party Deposit Contract If a 3 rd party has paid the cash deposit for the account, reference the deposit contract associated with the 3rd party.

Comments Enter any free-form comments about the non-cash deposit.

Account - Characteristics

The characteristics page contains information that describes miscellaneous information about the account. Use **Customer Management, Account** and navigate to the **Characteristics** tab to open this page.

Description of Page

> **Note:** You can only choose characteristic types defined as permissible on the account record. Refer to Setting Up Characteristic Types & Their Values for more information.

The following fields display:

Effective Date Indicate the date on which the characteristic value becomes effective. Note, the effective date defaults to the account's setup date (defined on the **Main** page).

Characteristic Type Indicate the type of characteristic.

Copyright © 2024, Oracle and/or its affiliates.

Characteristic Value Indicate the value of the characteristic.

Account - Alerts

The Alerts tab allows you to associate different types of alerts with an account. Once you associate an alert type with an account, it appears on the dashboard for the specified period. You can set an alert for any follow up action that needs to be taken for an account. For example, you can associate an alert type for setting a reminder for the due date of a bill payment for an account. This tab contains the following fields:

Field Name	Field Description
Alert Type	Used to specify the alert type for the account.
Start Date	Used to specify the date from when the alert type will be active for the account and displayed on the dashboard.
End Date	Used to specify the date till when the alert type will be active for the account and displayed on the dashboard.
	Note: If you want the alert type to be active for an indefinite period, leave this field blank. In this case, the active period will be the number of alert days defined for the alert type, which will be calculated from the specified start date.

Account - Account Identifiers

The **Account Identifiers** tab allows you to define the various types of identification associated with the account. For example, bank account number, account name, International Bank Account Number (IBAN), etc. This tab contains the following fields:

Field Name	Field Description
Primary ID	Used to indicate whether the account identifier is the primary identifier for the account.
Account Identifier Type	Used to specify the account identifier type.
Account Identifier	Used to specify the value for the account identifier type.

Note: If the **Account Identifier Required** option type of the **C1_ACCTINFO** feature configuration is set to **Y**, you have to specify primary identifier for the account. However, if you set the **Account Identifier Required** option type to **Y**, you will not be able to use the **Add Account and Start Service** feature available in the **Person** screen.

Account - Usage Amount/Counter

The **Usage Amount/Counter** zone allows you to search various amount and counter related information, such as average daily balance, average monthly balance, number of auto transfer transactions, and so on, of the account on a particular date and time. It contains the following two sections:

- **Search Criteria** - The **Search Criteria** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Effective From Date	Used to search amount or counter details of the account from a particular date onwards.	No
Effective To Date	Used to search amount or counter details of the account till a particular date.	No
Usage Amount	Used to search a particular type of amount details. The valid values are:	No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	• Combined Average Daily Balance • Combined Average Monthly Balance • Combined Balance • Combined Monthly Balance **Note:** You can add the required amount types in the **BAL_FLG** lookup field.	
Usage Counter	Used to search a particular type of counter details. The valid values are: • No. of ATM transactions • No. of Auto Transfer **Note:** You can add the required counter types in the **CNT_FLG** lookup field.	No
Usage Type	Used to indicate whether you want to search amount or counter details. The valid values are: • Usage Amount • Usage Counter	No

Note: If you search the amount or counter details without specifying any criteria, the system lists the amount and counter details of the account which are available on the current date.

- **Search Results** - On clicking the **Search** button, the search results appear based on the specified search criteria. The **Search Results** section contains the following columns:

Column Name	Column Description
Effective Date	Displays the date for which the amount or counter details are available.
Usage Type	Indicates whether the amount or counter details are available. The valid values are: • Usage Amount • Usage Counter
Usage Criteria	Indicates the type of amount or counter details.
Usage	Displays the amount or counter depending on the usage criteria.
Currency	Indicates the currency in which the amount is available. **Note:** The data does not appear in this column when the counter details are displayed.

Account - Contracts For A Debt Class

This is an information-only page that shows an account's contracts grouped by debt class.

Note: The source of debt class. An account's debt comes from its contract(s). Every contract has an contract type. Every contract type references a debt class.

To access this page, drill down on a debt class shown on **Customer Management, Account, Main**.

Description of Page

Copyright © 2024, Oracle and/or its affiliates.

The **Account ID** and the primary name of the account's main customer are displayed.

Debt Class is the debt class associated with the account's contract(s).

> **Note:** An account may have contracts that belong to other debt classes. Navigate to Account - Main Information to see a list of all debt classes associated with an account.

The **Contract** scroll contains one entry for each contract that belongs to the **Debt Class**. The following information is displayed for each contract.

- **Contract** (contract) **Info** is a concatenation of important details about the contract.
- **Current Balance** is the amount of debt the customer currently owes.
- **Payoff Balance** is the amount the customer would owe if they wanted to pay off their debt. Payoff balance is only displayed if it differs from current balance.

> **CAUTION:** If you do not understand the difference between payoff balance and current balance, refer to Current Amount versus Payoff Amount.

- **Contract Arrears** displays the age of the amounts that make up the current balance.
- The **Days Old** and **Arrears Amount** information shows the age of the contract's debt.

> **Note:** Information about debt balances by age is not applicable to open-item accounts.

Account Closure

You can set closing date for an account. However, the closing date cannot be earlier than the account's set up date, start date of any auto pay instruction defined on the account, and start date of any contract associated with the account. To enable the account closure feature, you need to create an algorithm using the **C1-CLSACCT** algorithm type and attach it to the **Account - Post Processing** system event of the division to which the account belongs.

This algorithm is invoked when the closing date is specified while adding and editing an account. It recalculates the closing date considering the grace days. The recalculated account closing date is then set as the end date of each contract associated with the account. In addition, the recalculated account closing date is set as the end date of the statement construct in which the account is included. If the **Update Auto Pay End Date** parameter is set to **Y**, the recalculated account closing date is set as the end date for each automatic payment instruction defined for the account. If there is any contract associated with the account which is in the **Active** status, the status of the contract is changed to **Pending Stop**.

Once these changes are done, a To Do is created using the specified To Do type and sent to the users with the specified To Do role to inform them about the recalculated account closing date.

This algorithm contains the following parameters:

- **Number of Grace Days** - Used to specify the number of grace days you want to add and accordingly recalculate the account closing date. This parameter is mandatory.
- **Grace Days Type (C - Calendar Days, B - Business Days)** - Used to indicate whether you want to use the calendar or business days while recalculating the account closing date. The business days are derived using the work calendar defined on the division. The valid values are:
  - C
  - B

  > **Note:** If you do not specify any value for this parameter, by default, it is set to **B**.
- **Update Auto Pay End Date (Y or N)** - Used to indicate whether you want to change the end date of each automatic payment instruction to the recalculated account closing date. The valid values are:
  - Y
  - N

Copyright © 2024, Oracle and/or its affiliates.

- **To Do Type** - Used to indicate the type of notification that you want to generate when the account closing date is recalculated.
- **To Do Role** - Used to specify the To Do role to indicate the users to whom you want to send the notification when the account closing date is recalculated.

AutoPay Add/Update, Delete, and View Through Inbound Web Service

Oracle Revenue Management and Billing enables you to define, edit, delete, and view auto pay instructions of an account through a web service. You can send the auto pay addition/modification, auto pay deletion, or auto pay view request from the external system through an inbound web service. The following three business services are shipped with the product to support different auto pay instruction operations:

Business Service	Description
C1-AddUpdateAutoPayService	Used to define or edit an auto pay instruction of an account. While sending the auto pay addition/modification request, you need to specify either of the following: • Account ID • Account Identifier Type, Account Identifier, and Division On processing the auto pay addition/modification request, the system first searches whether the auto pay instruction already exists for the account in the system. If so, the system edits the existing auto pay instruction of the account. If the auto pay instruction does not exist for the account, the system adds the auto pay instruction for the account.
	Note: The system searches for an auto pay instruction either using the auto pay ID or using the start date, auto pay source, auto pay route type, priority, and external account number combination.
	The system enables you to define or edit an auto pay instruction of an account through the web service when the **Rule Based Auto Pay** option is not selected for the account. The approval workflow process is supported through the web service when you create or edit one auto pay instruction for an account and not when you create or edit multiple auto pay instructions or split auto pay instruction for an account. You can also create split auto pay instructions for an account through the web service. However, you need to ensure that split auto pay instructions with the same start date, end date, and priority computes to 100% in the request. For more information, see C1-AddUpdateAutoPay Business Service on page 29
C1-DeleteAutoPayService	Used to delete an auto pay instruction of an account. While sending the auto pay deletion request, you need to specify either of the following: • Account ID • Account Identifier Type, Account Identifier, and Division On processing the auto pay deletion request, the system deletes an auto pay instruction of the account. If you specify the account details, the system deletes all auto pay instructions of the account. And, if you specify the auto pay ID, the system deletes the respective auto pay instruction of the account. Note that the system deletes an auto pay instruction of the account when it is not yet used in the system. For more information, see C1-DeleteAutoPay Business Service on page 45
C1-ViewAutoPayService	Used to view the auto pay instructions of an account. While sending the auto pay view request, you need to specify either of the following: • Account ID • Account Identifier Type, Account Identifier, and Division

Copyright © 2024, Oracle and/or its affiliates.

Business Service	Description
	You can specify various criteria (such as, start date and auto pay source) along with the account details to filter the list of auto pay instructions. For more information, see C1-ViewAutoPay Business Service on page 52

The system enables you to track the auto pay addition/modification, auto pay deletion, and auto pay view responses sent to the external system using the **Inbound Web Service History** feature.

We have not shipped any inbound web service with the product. You can create one inbound web service which supports all three operations - auto pay addition/modification, auto pay deletion, and auto pay view. Alternatively, you can create three inbound web services - one for auto pay addition/modification, another for auto pay deletion, and third for auto pay view.

You can create a SOAP or REST based inbound web service. If you want to send the request in the XML and/or JSON format, you need to ensure that you create a REST based inbound web service. However, if you want to send the request in the XML format, you can create a SOAP or REST based inbound web service.

Prerequisites

To create account auto pay instructions through an inbound web service, you need to do the following:

- Define a REST or SOAP based inbound web service to call the **Add Update Auto Pay (C1-AddUpdateAutoPay)** business service.
- Define the required account identifier types in the system.
- Define the users of the external system with appropriate authorization and/or access to the adjustment inbound web service in the system.
- Define the required auto pay source and auto pay route type in the system.
- Define the required characteristics to be stamped on the auto pay, where the characteristic type of those characteristics must be mapped with the auto pay route type.
- Define the required division in the system.
- Define an outbound message type (for example, C1-IWSHIST) with the following specifications:
  - **Outbound Message Business Object** - Web Service History (C1-WebServiceHistory)
  - **Priority** - Priority 10 -- Highest
- Define an external system and associate the above outbound message type (i.e. C1-IWSHIST) with the external system using the following specifications:
  - **Processing Method** - Batch
  - **Batch Control** - F1-FLUSH
- Create an algorithm using the **C1-IWSHIST** algorithm type and attach it to the **Web Service History** system event in the installation options.

> **Note:** You need to specify the above outbound message type (i.e. C1-IWSHIST), using which the outbound message should be created, in the **C1-IWSHIST** algorithm.

C1-AddUpdateAutoPay Business Service

The **Auto Pay Maintenance for Account (C1-AddUpdateAutoPayService)** business service enables you to add and update the autopay details of an account. You can call this business service from an external system through an inbound web service. If you want to send the adjustment creation request in the XML and/or JSON format, you need to ensure that you create a REST based inbound web service in Oracle Revenue Management and Billing. However, if you want to send the adjustment creation request in the XML format, you can create a SOAP or REST based inbound web service in Oracle Revenue Management and Billing.

To add and update the autopay details, you can either specify an account Id or account identifier type alongside the account number and division to the business service. In case you provide both (accountId and account identifier type), accountId is given precedence ahead of the account identifier type.

Copyright © 2024, Oracle and/or its affiliates.

You can specify the following mandatory parameters,

- **Start Date**
- **Auto Pay Source Code**
- **Auto Pay Route Type**
- **Priority**
- **External Account Number**

in the business service, as a request, if you want to search and retrieve a particular autopay record.

If the search parameters do not find an existing autopay record, the business service creates a new record for the account. It also executes the validation algorithm to validate the autopay details before creating a new record. Once autopay record is successfully created, the business service returns the response status back to the inbound web service in the form of **Success** or **Fail**. However, if the search parameters successfully locate an existing autopay record, the business service then performs an update operation on that autopay record, and returns an appropriate response back to the inbound web service.

Note: To perform a successful update, ensure that you provide entire information about the autopay Id in the C1-AddUpdateAutoPay Business Service on page 29.

AutoPay Add/Update Request in XML Format

The following code snippet illustrates how the tags should be nested while receiving the autopay add/update request in the XML format:

```
<C1-AddUpdateAutoPayService>
    <accountIdentifierType></accountIdentifierType>
    <accountNumber></accountNumber>
    <division></divisionCode>
    <accountId></accountId>
    <sourceSystem></sourceSystem>
    <c1ExternalTransactionId></externalTransactionId>
    <externalSourceId></externalSourceId>
    <accountautopaydata>
        <accountAutoPayId></accountAutoPayId>
        <startDate></startDate>
        <endDate></endDate>
        <autopaySource></autopaySource>
        <autopayRouteType></autopayRouteType>
        <percent></percent>
        <priorityNum></priorityNum>
        <expirationMonth></expirationMonth>
        <expirationYear></expirationYear>
        <externalAccountId></externalAccountId>
        <entityName></entityName>
        <autopayMaxWithdrawalAmount></autopayMaxWithdrawalAmount>
        <comments></comments>
        <autoPayMethod></autoPayMethod>
        <accountautopaymentchar>
            <characteristicType></characteristicType>
            <characteristicValue></characteristicValue>
        </accountautopaymentchar>
    </accountautopaydata>
</C1-AddUpdateAutoPayService>
```

AutoPay Add/Update Response in XML Format

The following code snippet illustrates how the tags are nested while sending the autopay add/update response in the XML format:

```
<C1-AddUpdateAutoPayService>
```

Copyright © 2024, Oracle and/or its affiliates.

```
    <responseStatus></responseStatus>
    <responseMessage></responseMessage>
    <accountIdentifierType></accountIdentifierType>
    <accountNumber></accountNumber>
    <division></divisionCode>
    <accountId</accountId>
    <responseStatus></responseStatus>
    <responseMessage></responseMessage>
    <sourceSystem></sourceSystem>
    <c1ExternalTransactionId></externalTransactionId>
    <externalSourceId></externalSourceId>
    <accountautopaydata>
        <accountAutoPayId></accountAutoPayId>
        <startDate></startDate>
        <endDate></endDate>
        <autopaySource></autopaySource>
        <autopayRouteType></autopayRouteType>
        <percent></percent>
        <priorityNum></priorityNum>
        <expirationMonth></expirationMonth>
        <expirationYear></expirationYear>
        <externalAccountId></externalAccountId>
        <entityName></entityName>
        <autopayMaxWithdrawalAmount></autopayMaxWithdrawalAmount>
        <comments></comments>
        <autoPayMethod></autoPayMethod>
        <accountautopaymentchar>
            <characteristicType></characteristicType>
            <characteristicValue></characteristicValue>
        </accountautopaymentchar>
        <accountAutoPayId>
            <startDate></startDate>
            <endDate></endDate>
            <autopaySource></autopaySource>
            <autopayRouteType></autopayRouteType>
            <percent></percent>
            <priorityNum></priorityNum>
            <expirationMonth></expirationMonth>
            <expirationYear></expirationYear>
            <externalAccountId></externalAccountId>
            <entityName></entityName>
            <autopayMaxWithdrawalAmount></autopayMaxWithdrawalAmount>
            <comments></comments>
            <autoPayMethod></autoPayMethod>
            <accountautopaymentchar>
                <characteristicType></characteristicType>
                <characteristicValue></characteristicValue>
            </accountautopaymentchar>
        </accountAutoPayId>
        <accountAutoPayId>
            <startDate></startDate>
            <endDate></endDate>
            <autopaySource></autopaySource>
            <autopayRouteType></autopayRouteType>
            <percent></percent>
            <priorityNum></priorityNum>
            <expirationMonth></expirationMonth>
            <expirationYear></expirationYear>
            <externalAccountId></externalAccountId>
            <entityName></entityName>
            <autopayMaxWithdrawalAmount></autopayMaxWithdrawalAmount>
            <comments></comments>
            <autoPayMethod></autoPayMethod>
        </accountAutoPayId>
```


Copyright © 2024, Oracle and/or its affiliates.

```
    </accountautopaydata>
</C1-AddUpdateAutoPayService>
```

AutoPay Add/Update Request in JSON Format

The following code snippet illustrates how the tags should be nested while receiving the autopay add/update request in the JSON format:

```
{
    -"C1-AddUpdateAutoPayService":
    {
        -"accountIdentifierType": -"",
        -"accountNumber": -"",
        -"division": -"",
        -"accountId": -"",
        -"sourceSystem": -"",
        -"c1ExternalTransactionId": -"",
        -"externalSourceId": -"",
        -"accountautopaydata":
        [
            {
                -"accountAutoPayId": -"",
                -"startDate": -"",
                -"endDate": -"",
                -"autopaySource": -"",
                -"autopayRouteType": -"",
                -"percent": -,
                -"priorityNum": -,
                -"expirationMonth": -"",
                -"expirationYear": -"",
                -"externalAccountId": -"",
                -"autopayMaxWithdrawalAmount": -,
                -"entityName": -"",
                -"comments": -"",
                -"autoPayMethod": -"",
                -"accountautopaymentchar":
                [
                    {
                        -"characteristicType": -"",
                        -"characteristicValue": -""
                    -}
                -]
            -}
            {
                -"accountAutoPayId": -"",
                -"startDate": -"",
                -"endDate": -"",
                -"autopaySource": -"",
                -"autopayRouteType": -"",
                -"percent": -,
                -"priorityNum": -,
                -"expirationMonth": -"",
                -"expirationYear": -"",
                -"externalAccountId": -"",
                -"autopayMaxWithdrawalAmount": -,
                -"entityName": -"",
                -"comments": -"",
                -"autoPayMethod": -"",
                -"accountautopaymentchar":
                [
                    {
                        -"characteristicType": -"",
                        -"characteristicValue": -""
```

Copyright © 2024, Oracle and/or its affiliates.

```
                    -}
                -]
            -}
        -]
    -}
}
```

AutoPay Add/Update Response in JSON Format

The following code snippet illustrates how the tags are nested while sending the autopay add/update response in the JSON format:

```
{
    -"C1-AddUpdateAutoPayService":
    {
        -"responseStatus": -"",
        -"responseMessage": -"",
        -"accountIdentifierType": -"",
        -"accountNumber": -"",
        -"division": -"",
        -"accountId": -"",
        -"sourceSystem": -"",
        -"c1ExternalTransactionId": -"",
        -"externalSourceId": -"",
        -"accountautopaydata": -""
       [
            {
                -"accountAutoPayId": -"",
                -"startDate": -"",
                -"endDate": -"",
                -"autopaySource": -"",
                -"autopayRouteType": -"",
                -"percent": -"",
                -"priorityNum": -"",
                -"expirationMonth": -"",
                -"expirationYear": -"",
                -"externalAccountId": -"",
                -"autopayMaxWithdrawalAmount": -"",
                -"entityName": -"",
                -"comments": -"",
                -"autoPayMethod": -"",
                -"accountautopaymentchar": -""
                [
                    {
                        -"characteristicType": -"",
                        -"characteristicValue": -""
                    -}
                -]
            -},
            {
                -"accountAutoPayId": -"",
                -"startDate": -"",
                -"endDate": -"",
                -"autopaySource": -"",
                -"autopayRouteType": -"",
                -"percent": -"",
                -"priorityNum": -"",
                -"expirationMonth": -"",
                -"expirationYear": -"",
                -"externalAccountId": -"",
                -"autopayMaxWithdrawalAmount": -"",
                -"entityName": -"",
                -"comments": -"",
```

Copyright © 2024, Oracle and/or its affiliates.

```
                    -"autoPayMethod": -"",
                    -"accountautopaymentchar": -""
                    [
                        {
                            -"characteristicType": -"",
                            -"characteristicValue": -""
                        -}
                    -]
                -},
            -]
        -}
}
```

AutoPay Add/Update Request and Response - Tags in XML Format

Note: We recommend you to refer the AutoPay Add/Update Request in XML Format on page 30 and AutoPay Add/Update Response in XML Format on page 30 topics in parallel while understanding the below mentioned tags. This will help you to understand how the tags are nested in the XML format.

Before calling the **C1-AddUpdateAutoPayService** business service through an inbound web service, you need to ensure that the autopay add/update request contains the following tags:

Tag Name	Tag Description	Mandatory (Yes or No)
accountIdentifierType	Used to specify the account identifier type. **Note:** You must specify an account identifier type which is already defined in the system.	Yes (Conditional) **Note:** This data is required while adding or updating the autopay details of an account.
accountNumber	Used to indicate the account for which you want to create or update the auto pay information.	Yes (Conditional) **Note:** This data is required when the **accountIdentifierType** is specified.
division	Used to indicate the division to which the account belongs. **Note:** You must specify the division code which is already defined in the system.	Yes (Conditional) **Note:** This data is required while adding or updating the autopay details of an account.
accountId	Used to indicate the account for which the auto pay details are added or updated. **Note:** You must specify the accountId which is already defined in the system.	Yes (Conditional) **Note:** This data is required if **accountIdentifierType** and **accountNumber** is not specified.
sourceSystem	Used to indicate the external system from where the autopay information is received.	No
c1ExternalTransactionId	Used to indicate the transaction in the external source system from where the auto pay information is received.	No

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Tag Description	Mandatory (Yes or No)
externalSourceId	Used to specify the external source system Id.	No
accountAutoPayData	Used to specify a list of autopay details for an account.	No
accountAutoPayId	Used to specify the auto pay Id for an account. **Note:** The accountAutoPayId is provided as an input parameter whenever the auto pay details are updated or deleted for an account.	No
startDate	Used to indicate the date from when the auto pay option is effective. **Note:** The startDate cannot not be later than the endDate.	Yes
endDate	Used to indicate the date till when the auto pay option is effective. **Note:** The startDate cannot not be earlier than the endDate.	No
autopaySource	Used to indicate the source of funds used to make the automatic payment. For example, debit card or credit card. **Note:** You must specify the autopaySource which is already defined in the system.	Yes
autopayRouteType	Used to indicate when and how automatic payment request of the account is routed to a financial institution. **Note:** You must specify an autopay route type which is already defined in the system.	Yes
percent	Used to indicate the percentage in which the payment must be split between two or more autopay sources having the same priority.	No
priorityNum	Used to indicate the priority in which the auto pay instruction should be considered when multiple autopay instructions are effective on the bill's due date. **Note:** You must specify the priorityNum which is already defined in the system without the occurrence of a decimal.	Yes
expirationMonth	Used to specify the month when the autopay source will expire. **Note:** The expiration month should not be earlier than the month of the start date.	No
expirationYear	Used to specify the year when the autopay source will expire. **Note:** The expiration year should not be earlier than the year of the start date.	No

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Tag Description	Mandatory (Yes or No)
externalAccountId	Used to indicate the account in the external system for which the autopay details should be added or updated.	Yes
entityName	Used to specify the name of the person.	No
autopayMax WithdrawalAmount	Used to specify the maximum amount that can be automatically debited from the autopay account. It is used to set the limit on the withdrawal amount. **Note:** The autopay max withdrawal amount must be a positive number with or without a decimal. It should not be less than zero.	No
comments	Used to specify additional information about the autopay instruction.	No
autoPayMethod	Used to indicate how you want to process the automatic payment request. The valid values are: • **C1DD** - Indicates that you want to directly debit the main customer's bank account. • **C1PA** - Indicates that you want to send a payment advice to the main customer. **Note:** You must specify the autopay method which is already defined in the system.	No
accountautopaymentchar	Used to specify a list of autopay characteristics for an account.	No
characteristicType	You must specify a characteristic type where the characteristic entity is set to **Account**. **Note:** You must specify the characteristic type which is already defined in the system.	No
characteristicValue	Used to specify the value for the characteristic type.	No

The following table lists and describes the tags which appear in the autopay add/update response in the XML format:

Tag Name	Tag Description
responseStatus	Indicates whether the autopay add/update request is successfully processed or not. The valid values are: • Success • Fail
responseMessage	Indicates an error that occurred while adding/updating an autopay request. **Note:** This tag appears when the value in the **responseStatus** tag is set to **Fail**.

Related Topics

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
Sample AutoPay Add/Update Request and Response in XML Format	Sample AutoPay Add/Update Request and Response in XML Format on page 40

AutoPay Add/Update Request and Response - Attribute/Value Pairs in JSON Format

Note: We recommend you to refer the AutoPay Add/Update Request in JSON Format on page 32 andAutoPay Add/Update Response in JSON Format on page 33 topics in parallel while understanding the below mentioned attributes. This will help you to understand how the attribute/value pairs are structured in the JSON format.

Before calling the **C1-AddUpdateAutoPayService** business service, you need to ensure that the autopay add/update service contains the following attributes:

Attribute Name	Attribute Description	Mandatory (Yes or No)
accountIdentifierType	Used to specify the account identifier type. **Note:** You must specify an account identifier type which is already defined in the system.	Yes (Conditional) **Note:** This data is required while adding or updating the autopay details of an account.
accountNumber	Used to indicate the account for which you want to add or update the autopay information.	Yes (Conditional) **Note:** This data is required when the **accountIdentifierType** is specified.
division	Used to indicate the division to which the account belongs. **Note:** You must specify the division code which is already defined in the system.	Yes (Conditional) **Note:** This data is required while adding or updating the autopay details of an account.
accountId	Used to indicate the account for which the autopay details are added or updated. **Note:** You must specify the accountId which is already defined in the system.	Yes (Conditional) **Note:** This data is required if **accountIdentifierType** and **accountNumber** is not specified.
sourceSystem	Used to indicate the external system from where the autopay information is received.	No
c1ExternalTransactionId	Used to indicate the transaction in the external source system from where the autopay information is received.	No
externalSourceId	Used to specify the externalSourceId.	No
accountAutoPayData	Used to specify a list of autopay details for an account.	No
accountAutoPayId	Used to specify the autopayId for an account.	No

Copyright © 2024, Oracle and/or its affiliates.

Attribute Name	Attribute Description	Mandatory (Yes or No)
	Note: The accountAutoPayId is provided as an input parameter whenever the autopay details are updated or deleted for an account.	
startDate	Used to indicate the date from when the autopay option is effective. **Note:** The startDate cannot not be later than the endDate.	Yes
endDate	Used to indicate the date till when the autopay option is effective. **Note:** The startDate cannot not be earlier than the endDate.	No
autopaySource	Used to indicate the source of funds used to make the automatic payment. For example, debit card or credit card. **Note:** You must specify the autopaySource which is already defined in the system.	Yes
autopayRouteType	Used to indicate when and how automatic payment request of the account is routed to a financial institution. **Note:** You must specify an auto pay route type which is already defined in the system.	Yes
percent	Used to indicate the percentage in which the payment must be split between two or more auto pay sources having the same priority.	No
priorityNum	Used to indicate the priority in which the autopay instruction should be considered when multiple auto pay instructions are effective on the bill's due date. **Note:** You must specify the priorityNum which is already defined in the system without the occurrence of a decimal.	Yes
expirationMonth	Used to specify the month when the autopay source will expire. **Note:** The expiration month should not be earlier than the month of the start date.	No
expirationYear	Used to specify the year when the autopay source will expire. **Note:** The expiration year should not be earlier than the year of the start date.	No
externalAccountId	Used to indicate the account in the external system for which the autopay details should be added or updated.	Yes
entityName	Used to specify the name of the person.	No

Copyright © 2024, Oracle and/or its affiliates.

Attribute Name	Attribute Description	Mandatory (Yes or No)
autopayMax WithdrawalAmount	Used to specify the maximum amount that can be automatically debited from the autopay account. It is used to set the limit on the withdrawal amount. **Note:** The autopay max withdrawal amount must be a positive number with or without a decimal. It should not be less than zero.	No
comments	Used to specify additional information about the autopay instruction.	No
autoPayMethod	Used to indicate how you want to process the automatic payment request. The valid values are: • **C1DD** - Indicates that you want to directly debit the main customer's bank account. • **C1PA** - Indicates that you want to send a payment advice to the main customer. **Note:** You must specify the autopay method which is already defined in the system.	No
accountautopaymentchar	Used to specify a list of autopay characteristics for an account.	No
characteristicType	You must specify a characteristic type where the characteristic entity is set to **Account**. **Note:** You must specify the characteristic type which is already defined in the system.	No
characteristicValue	Used to specify the value for the characteristic type.	No

The following table lists and describes the attributes which appear in the autopay add/update response in the JSON format:

Attribute Name	Attribute Description
responseStatus	Indicates whether the autopay add/update request is successfully processed or not. The valid values are: • Success • Fail
responseMessage	Indicates an error that occurred while conducting an add/update of an autopay request. **Note:** This attribute appears when the value in the **responseStatus** tag is set to **Fail**.

Related Topics

For more information on...	See...
Sample AutoPay Add/Update Request and Response in JSON Format	Sample AutoPay Add/Update Request and Response in JSON Format on page 42

Copyright © 2024, Oracle and/or its affiliates.

Sample AutoPay Add/Update Request and Response in XML Format

Using the below example, you can add/update an autopay request for the 7652701564 account.

Sample AutoPay Add/Update Request in XML Format

```
<C1-AddUpdateAutoPayService>
    <accountIdentifierType></accountIdentifierType>
    <accountNumber></accountNumber>
    <division></divisionCode>
    <accountId>7652701564</accountId>
    <sourceSystem></sourceSystem>
    <c1ExternalTransactionId></externalTransactionId>
    <externalSourceId></externalSourceId>
    <accountautopaydata>
        <accountAutoPayId></accountAutoPayId>
        <startDate>2011-12-07</startDate>
        <endDate>2014-12-07</endDate>
        <autopaySource>CITIBANKCHK</autopaySource>
        <autopayRouteType>ACH2</autopayRouteType>
        <percent>35</percent>
        <priorityNum>4</priorityNum>
        <expirationMonth>01</expirationMonth>
        <expirationYear>2020</expirationYear>
        <externalAccountId>444444</externalAccountId>
        <autopayMaxWithdrawalAmount>100</autopayMaxWithdrawalAmount>
        <entityName>RJDas</entityName>
        <comments>Testing</comments>
        <autoPayMethod>C1DD</autoPayMethod>
        <accountautopaymentchar>
            <characteristicType>CMSIFSSF</characteristicType>
            <characteristicValue>MN</characteristicValue>
        </accountautopaymentchar>
    </accountautopaydata>
    <accountautopaydata>
        <accountAutoPayId></accountAutoPayId>
        <startDate>2011-12-07</startDate>
        <endDate>2014-12-07</endDate>
        <autopaySource>CITIBANKCHK</autopaySource>
        <autopayRouteType>NEFT</autopayRouteType>
        <percent>15</percent>
        <priorityNum>4</priorityNum>
        <expirationMonth>02</expirationMonth>
        <expirationYear>2020</expirationYear>
        <externalAccountId>789654</externalAccountId>
        <autopayMaxWithdrawalAmount>10</autopayMaxWithdrawalAmount>
        <entityName>GKDas</entityName>
        <comments>Testing</comments>
        <autoPayMethod>C1DD</autoPayMethod>
        <accountautopaymentchar>
            <characteristicType>CMSIFSSF</characteristicType>
            <characteristicValue>MN</characteristicValue>
        </accountautopaymentchar>
    </accountautopaydata>
    <accountautopaydata>
        <accountAutoPayId></accountAutoPayId>
        <startDate>2011-12-07</startDate>
        <endDate>2014-12-07</endDate>
        <autopaySource>CITIBANKCHK</autopaySource>
        <autopayRouteType>ACH</autopayRouteType>
        <percent>50</percent>
        <priorityNum>4</priorityNum>
        <expirationMonth>01</expirationMonth>
        <expirationYear>2019</expirationYear>
```

Copyright © 2024, Oracle and/or its affiliates.

```
        <externalAccountId>456321</externalAccountId>
        <autopayMaxWithdrawalAmount>100</autopayMaxWithdrawalAmount>
        <entityName>MKDas</entityName>
        <comments>Testing</comments>
        <autoPayMethod>C1DD</autoPayMethod>
    </accountautopaydata>
</C1-AddUpdateAutoPayService>
```


Sample AutoPay Add/Update Response in XML Format


```
<C1-AddUpdateAutoPayService>
    <responseStatus>Success</responseStatus>
    <responseMessage>AutoPay Id Added/Updated Successfully</responseMessage>
    <accountIdentifierType></accountIdentifierType>
    <accountNumber></accountNumber>
    <division></divisionCode>
    <accountId>7652701564</accountId>
    <sourceSystem></sourceSystem>
    <c1ExternalTransactionId></externalTransactionId>
    <externalSourceId></externalSourceId>
    <accountautopaydata>
        <accountAutoPayId></accountAutoPayId>
        <startDate>2011-12-07</startDate>
        <endDate>2014-12-07</endDate>
        <autopaySource>CITIBANKCHK</autopaySource>
        <autopayRouteType>ACH2</autopayRouteType>
        <percent>35</percent>
        <priorityNum>4</priorityNum>
        <expirationMonth>01</expirationMonth>
        <expirationYear>2020</expirationYear>
        <externalAccountId>444444</externalAccountId>
        <autopayMaxWithdrawalAmount>100</autopayMaxWithdrawalAmount>
        <entityName>RJDas</entityName>
        <comments>Testing</comments>
        <autoPayMethod>C1DD</autoPayMethod>
        <accountautopaymentchar>
            <characteristicType>CMSIFSSF</characteristicType>
            <characteristicValue>MN</characteristicValue>
        </accountautopaymentchar>
    </accountautopaydata>
    <accountautopaydata>
        <accountAutoPayId></accountAutoPayId>
        <startDate>2011-12-07</startDate>
        <endDate>2014-12-07</endDate>
        <autopaySource>CITIBANKCHK</autopaySource>
        <autopayRouteType>NEFT</autopayRouteType>
        <percent>15</percent>
        <priorityNum>4</priorityNum>
        <expirationMonth>02</expirationMonth>
        <expirationYear>2020</expirationYear>
        <externalAccountId>789654</externalAccountId>
        <autopayMaxWithdrawalAmount>10</autopayMaxWithdrawalAmount>
        <entityName>GKDas</entityName>
        <comments>Testing</comments>
        <autoPayMethod>C1DD</autoPayMethod>
        <accountautopaymentchar>
            <characteristicType>CMSIFSSF</characteristicType>
            <characteristicValue>MN</characteristicValue>
        </accountautopaymentchar>
    </accountautopaydata>
    <accountautopaydata>
        <accountAutoPayId></accountAutoPayId>
        <startDate>2011-12-07</startDate>
```


Copyright © 2024, Oracle and/or its affiliates.

```
        <endDate>2014-12-07</endDate>
        <autopaySource>CITIBANKCHK</autopaySource>
        <autopayRouteType>ACH</autopayRouteType>
        <percent>50</percent>
        <priorityNum>4</priorityNum>
        <expirationMonth>01</expirationMonth>
        <expirationYear>2019</expirationYear>
        <externalAccountId>456321</externalAccountId>
        <autopayMaxWithdrawalAmount>100</autopayMaxWithdrawalAmount>
        <entityName>MKDas</entityName>
        <comments>Testing</comments>
        <autoPayMethod>C1DD</autoPayMethod>
    </accountautopaydata>
</C1-AddUpdateAutoPayService>
```

Note: We have given the above sample to illustrate how the tags should be nested while sending an auto pay add or update request from the external system. However, you should not seamlessly use this sample in your environment because the required pre-requisite data may not be available in your environment. In such case, the system will give erroneous results.

Sample AutoPay Add/Update Request and Response in JSON Format

Using the below example, you can add/update an autopay request for the 7652701564 account.

Sample AutoPay Add/Update Request in JSON Format

```
{
    -"C1-AddUpdateAutoPayService":
    {
        -"accountIdentifierType": -"",
        -"accountNumber": -"",
        -"division": -"",
        -"accountId": -"7652701564",
        -"sourceSystem": -"",
        -"c1ExternalTransactionId": -"",
        -"externalSourceId": -"",
        -"accountautopaydata":
        [
            {
                -"accountAutoPayId": -"",
                -"startDate": -"2011-12-07",
                -"endDate": -"2014-12-07",
                -"autopaySource": -"CITIBANKCHK",
                -"autopayRouteType": -"ACH2",
                -"percent": 35,
                -"priorityNum": 4,
                -"expirationMonth": -"01",
                -"expirationYear": -"2020",
                -"externalAccountId": -"444444",
                -"autopayMaxWithdrawalAmount": 100,
                -"entityName": -"Raj",
                -"comments": -"Testing",
                -"autoPayMethod": -"C1DD",
                -"accountautopaymentchar":
                [
                    {
                        -"characteristicType": -"CMSIFSSF",
                        -"characteristicValue": -"MN"
                    -}
                -]
            -}
            {
                -"accountAutoPayId": -"",
```

Copyright © 2024, Oracle and/or its affiliates.

```
                -"startDate": -"2011-12-07",
                -"endDate": -"2014-12-07",
                -"autopaySource": -"CITIBANKCHK",
                -"autopayRouteType": -"NEFT",
                -"percent": 15,
                -"priorityNum": 4,
                -"expirationMonth": -"02",
                -"expirationYear": -"2020",
                -"externalAccountId": -"789654",
                -"autopayMaxWithdrawalAmount": 10,
                -"entityName": -"ABCL",
                -"comments": -"For Testing",
                -"autoPayMethod": -"C1DD",
                -"accountautopaymentchar":
                [
                    {
                        -"characteristicType": -"CMSIFSSF",
                        -"characteristicValue": -"MN"
                    -}
                -]
            -}
            {
                -"accountAutoPayId": -"",
                -"startDate": -"2011-12-07",
                -"endDate": -"2014-12-07",
                -"autopaySource": -"CITIBANKCHK",
                -"autopayRouteType": -"ACH",
                -"percent": 50,
                -"priorityNum": 4,
                -"expirationMonth": -"01",
                -"expirationYear": -"2019",
                -"externalAccountId": -"456321",
                -"autopayMaxWithdrawalAmount": 100,
                -"entityName": -"PQRS",
                -"comments": -"For Testing",
                -"autoPayMethod": -"C1DD",
             -}
        -]
    -}
}
```


Sample AutoPay Add/Update Response in JSON Format


```
{
    -"C1-AddUpdateAutoPayService":
    {
        -"responseStatus": -"Success",
        -"responseMessage": -"AutoPay Id Added -/ Updated Successfully",
        -"accountIdentifierType": -"",
        -"accountNumber": -"",
        -"division": -"",
        -"accountId": -"7652701564",
        -"sourceSystem": -"",
        -"c1ExternalTransactionId": -"",
        -"externalSourceId": -"",
        -"accountautopaydata":
       [
            {
                -"accountAutoPayId": -"",
                -"startDate": -"2011-12-07",
                -"endDate": -"2014-12-07",
                -"autopaySource": -"CITIBANKCHK",
                -"autopayRouteType": -"ACH2",
```


Copyright © 2024, Oracle and/or its affiliates.

```
                -"percent": 35,
                -"priorityNum": 4,
                -"expirationMonth": -"01",
                -"expirationYear": -"2020",
                -"externalAccountId": -"444444",
                -"autopayMaxWithdrawalAmount": 100,
                -"entityName": -"Raj",
                -"comments": -"For Testing Purposes",
                -"autoPayMethod": -"C1DD",
                -"accountautopaymentchar":
                [
                    {
                        -"characteristicType": -"CMSIFSSF",
                        -"characteristicValue": -"MN"
                    -}
                -]
            -},
            {
                -"accountAutoPayId": -"",
                -"startDate": -"2011-12-07",
                -"endDate": -"2014-12-07",
                -"autopaySource": -"CITIBANKCHK",
                -"autopayRouteType": -"NEFT",
                -"percent": 15,
                -"priorityNum": 4,
                -"expirationMonth": -"02",
                -"expirationYear": -"2020",
                -"externalAccountId": -"789654",
                -"autopayMaxWithdrawalAmount": 10,
                -"entityName": -"GKDas",
                -"comments": -"Test",
                -"autoPayMethod": -"C1DD",
                -"accountautopaymentchar":
                [
                    {
                        -"characteristicType": -"CMSIFSSF",
                        -"characteristicValue": -"MN"
                    -}
                -]
            -},
            {
                -"accountAutoPayId": -"",
                -"startDate": -"2011-12-07",
                -"endDate": -"2014-12-07",
                -"autopaySource": -"CITIBANKCHK",
                -"autopayRouteType": -"ACH",
                -"percent": 50,
                -"priorityNum": 4,
                -"expirationMonth": -"01",
                -"expirationYear": -"2019",
                -"externalAccountId": -"456321",
                -"autopayMaxWithdrawalAmount": 100,
                -"entityName": -"Rajesh",
                -"comments": -"Testing",
                -"autoPayMethod": -"C1DD",
            -},
        -]
    -}
}
```


Copyright © 2024, Oracle and/or its affiliates.

Note: We have given the above sample to illustrate how the attribute value pairs should be structured while sending an auto pay add or update request from the external system. However, you should not seamlessly use this sample in your environment because the required pre-requisite data may not be available in your environment. In such case, the system will give erroneous results.

C1-DeleteAutoPay Business Service

The **Delete Auto Pay From an Account (C1-DeleteAutoPayService)** inbound web service is used for deleting the auto pay details from the account, in the Oracle Revenue Management and Billing application from an external system. You can identify the auto pay details of an account by providing an accountId or account identifier type along with account number and division to the inbound web service. If both, accountId and account identifier type are provided, accountId is given precedence ahead of the account identifier type.

You cannot directly delete an account autopay Id without undergoing the referential integrity check. The system displays the referential integrity error if in case the account autopay Id has a reference with some other database table. You can delete all of the non-referred account autopay Ids on an account if you have provided an accountId or account identifier type along with account number and division as auto pay details to the inbound web service. If an account autopay Id is provided in the auto pay details, system grants highest priority to the account autopay Id and further deletes the auto pay details for an account which is not referenced anywhere in the system.

Once the auto pay details are successfully deleted for an account the inbound service returns the following response:

- accountAutoPayId: {List of the deleted autopay Ids}
- responseStatus: {Success}
- responseMessage: AutoPay Deleted Successfully of the account Id {xzxzxzxz}

If an error has occurred while deleting the account autopay Ids, the inbound service returns the following response:

- responseMessage: {error message is returned}

AutoPay Deletion Request in XML Format

The following code snippet illustrates how the tags should be nested while receiving an autopay deletion request in the XML format:

Note: You can delete an autopay request either for an accountId or autopayId.

Deleting the autopay information for an AccountId

```
<C1-DeleteAutoPayService>
    <accountId></accountId>
    <accountIdentifierType></accountIdentifierType>
    <accountNumber></accountNumber>
    <division></division>
    <sourceSystem></sourceSystem>
    <c1ExternalTransactionId></c1ExternalTransactionId>
    <externalSourceId></externalSourceId>
</C1-DeleteAutoPayService>
```

Deleting the autopay information for an AutoPayId

```
<C1-DeleteAutoPayService>
    <accountId></accountId>
    <accountIdentifierType></accountIdentifierType>
    <accountNumber></accountNumber>
    <division></division>
    <sourceSystem></sourceSystem>
    <c1ExternalTransactionId></c1ExternalTransactionId>
    <externalSourceId></externalSourceId>
    <accountautopaydata>
        <accountAutoPayId></accountAutoPayId>
```

Copyright © 2024, Oracle and/or its affiliates.

```
    </accountautopaydata>
</C1-DeleteAutoPayService>
```

AutoPay Deletion Response in XML Format

The following code snippet illustrates how the tags should be nested while sending the autopay deletion response in the XML format:

Note: You can delete an autopay request either for an accountId or autopayId.

Deleting the Autopay information for an AccountId or AutoPayId

```
<C1-DeleteAutoPayService>
    <accountautopayid>
        <AccountAutoPayId></AccountAutoPayId>
    </accountautopayid>
    <responseStatus></responseStatus>
    <responseMessage></responseMessage>
</C1-DeleteAutoPayService>
```

AutoPay Deletion Request in JSON Format

The following code snippet illustrates how the tags should be nested while receiving the autopay deletion request in the JSON format:

Note: You can delete an autopay request either for an accountId or autopayId.

Deleting the autopay information for an AccountId

```
{
    -"C1-DeleteAutoPayService":
    {
        -"accountId": -"",
        -"accountIdentifierType": -"",
        -"accountNumber": -"",
        -"division": -"",
        -"sourceSystem": -"",
        -"c1ExternalTransactionId": -"",
        -"externalSourceId": -"",
    -}
}
```

Deleting the autopay information for an AutoPayId

```
{
    -"C1-DeleteAutoPayService":
   {
        -"accountId": -"",
        -"accountIdentifierType": -"",
        -"accountNumber": -"",
        -"division": -"",
        -"sourceSystem": -"",
        -"c1ExternalTransactionId": -"",
        -"externalSourceId": -"",
        -"accountautopaydata":
      [
            {
                -"accountAutoPayId": -""
            -}
         -]
    -}
```

Copyright © 2024, Oracle and/or its affiliates.

```
}
```

AutoPay Deletion Response in JSON Format

The following code snippet illustrates how the tags should be nested while sending the autopay deletion response in the JSON format:

Note: You can delete an autopay request either for an accountId or autopayId.

Deleting the Autopay information for an AccountId

```
{
    -"C1-DeleteAutoPayService":
    {
        {
            -"AccountAutoPayId": -"8889143891,12333344554"
        -}
        -"responseStatus": -"Success",
        -"responseMessage": -"AutoPayIds are deleted successfully"
    -}
}
```

Deleting the Autopay information for an AutoPayId

```
{
    -"C1-DeleteAutoPayService":
    {
        {
            -"AccountAutoPayId": -"8889143136"
        -}
        -"responseStatus": -"Success",
        -"responseMessage": -"AutoPayId is deleted successfully"
    -}
}
```

AutoPay Deletion Request and Response - Tags in XML Format

Note: We recommend you to refer the AutoPay Deletion Request in XML Format on page 45 and AutoPay Deletion Response in XML Format on page 46 topics in parallel while understanding the below mentioned tags. This will help you to understand how the tags are nested in the XML format.

Before calling the **C1-DeleteAutoPayService** business service, you need to ensure that the autopay deletion request contains the following tags:

Tag Name	Tag Description	Mandatory (Yes or No)
accountIdentifierType	Used to specify the account identifier type.	Yes (Conditional)
	Note: You must specify an account identifier type which is already defined in the system.	**Note:** This data is required while deleting the autopay details of an account.
accountNumber	Used to indicate the account for which you want to create or update the autopay information.	Yes (Conditional)
		Note: This data is required when the **accountIdentifierType** is specified.

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Tag Description	Mandatory (Yes or No)
division	Used to indicate the division to which the account belongs. **Note:** You must specify the division code which is already defined in the system.	Yes (Conditional) **Note:** This data is required while adding or updating the autopay details of an account.
accountId	Used to indicate the account for which the autopay details are added or updated. **Note:** You must specify the accountId which is already defined in the system.	Yes (Conditional) **Note:** This data is required if **accountIdentifierType** and **accountNumber** is not specified.
sourceSystem	Used to indicate the external system from where the autopay information is received.	No
c1ExternalTransactionId	Used to indicate the transaction in the external source system from where the autopay information is received.	No
externalSourceId	Used to specify the external source system Id.	No
accountautopaydata	Used to specify a list of autopay details for an account.	No
accountAutoPayId	Used to specify the autopay Id for an account. **Note:** The accountAutoPayId is provided as an input parameter whenever the auto pay details are either updated or deleted for an account.	No

The following table lists and describes the tags which appear in the autopay deletion response in the XML format:

Tag Name	Tag Description
accountautopayid	Displays a list of account autopayIds which are deleted through an autopay deletion request.
responseStatus	Indicates whether the autopay deletion request is successfully processed or not. The valid values are: • Success • Fail
responseMessage	Indicates an error that occurred while deleting the autopayIds. **Note:** This tag appears when the value in the **responseStatus** tag is set to **Fail**.

Related Topics

For more information on...	See...
Sample AutoPay Deletion Request and Response in XML Format	Sample AutoPay Deletion Request and Response in XML Format on page 50

Copyright © 2024, Oracle and/or its affiliates.

AutoPay Deletion Request and Response - Attribute/Value Pairs in JSON Format

Note: We recommend you to refer the AutoPay Deletion Request in JSON Format on page 46 and AutoPay Deletion Response in JSON Format on page 47 topics in parallel while understanding the below mentioned attributes. This will help you to understand how the attribute/value pairs are structured in the JSON format.

Before calling the **C1-DeleteAutoPayService** business service, you need to ensure that the autopay deletion request contains the following attributes:

Attribute Name	Attribute Description	Mandatory (Yes or No)
accountIdentifierType	Used to specify the account identifier type. **Note:** You must specify an account identifier type which is already defined in the system.	Yes (Conditional) **Note:** This data is required while deleting the auto pay details of an account.
accountNumber	Used to indicate the account for which you want to create or update the autopay information.	Yes (Conditional) **Note:** This data is required when the **accountIdentifierType** is specified.
division	Used to indicate the division to which the account belongs. **Note:** You must specify the division code which is already defined in the system.	Yes (Conditional) **Note:** This data is required while adding or updating the auto pay details of an account.
accountId	Used to indicate the account for which the auto pay details are added or updated. **Note:** You must specify the accountId which is already defined in the system.	Yes (Conditional) **Note:** This data is required if **accountIdentifierType** and **accountNumber** is not specified.
sourceSystem	Used to indicate the external system from where the auto pay information is received.	No
c1ExternalTransactionId	Used to indicate the transaction in the external source system from where the auto pay information is received.	No
externalSourceId	Used to specify the externalSourceId.	No
accountautopaydata	Used to specify a list of autopay details for an account.	No
accountAutoPayId	Used to specify the autopayId for an account. **Note:** The accountAutoPayId is provided as an input parameter whenever the auto pay details are either updated or deleted for an account.	No

The following table lists and describes the attributes which appear in the autopay deletion response in the JSON format:

Copyright © 2024, Oracle and/or its affiliates.

Attribute Name	Attribute Description
accountautopayid	Displays a list of account autopayIds which are deleted through an autopay deletion request.
responseStatus	Indicates whether the autopay deletion request is successfully processed or not. The valid values are: • Success • Fail
responseMessage	Indicates an error that occurred while adding/updating an autopay request.
	Note: This attribute appears when the value in the **responseStatus** attribute is set to **Fail**.

Related Topics

For more information on...	See...
Sample AutoPay Deletion Request and Response in JSON Format	AutoPay Deletion Request in JSON Format on page 46

Sample AutoPay Deletion Request and Response in XML Format

Using the below example, you can delete an autopay request which has an incorrect data.

Sample AutoPay Deletion Request for an AccountId in XML Format

```
<C1-DeleteAutoPayService>
    <accountId>8889143815</accountId>
    <accountIdentifierType>ACCTNUM</accountIdentifierType>
    <accountNumber>2345678000</accountNumber>
    <division>D0301</division>
    <sourceSystem>DS01</sourceSystem>
    <c1ExternalTransactionId>787878</c1ExternalTransactionId>
    <externalSourceId>2240</externalSourceId>
</C1-DeleteAutoPayService>
```

Sample AutoPay Deletion Request for an AutoPayId in JSON Format

```
<C1-DeleteAutoPayService>
    <accountId>8889143815</accountId>
    <accountIdentifierType>ACCTNUM</accountIdentifierType>
    <accountNumber>2345679000</accountNumber>
    <division>D0401</division>
    <sourceSystem>ES01</sourceSystem>
    <c1ExternalTransactionId>989898</c1ExternalTransactionId>
    <externalSourceId>2250</externalSourceId>
    <accountautopaydata>
        <accountAutoPayId>7779143568</accountAutoPayId>
    </accountautopaydata>
</C1-DeleteAutoPayService>
```

Sample AutoPay Deletion Response for an AccountId in XML Format

```
<C1-DeleteAutoPayService>
    <accountautopayid>
        <AccountAutoPayId>8889143891,12333344554</AccountAutoPayId>
    </accountautopayid>
    <responseStatus>Success</responseStatus>
```

Copyright © 2024, Oracle and/or its affiliates.

```
    <responseMessage>AutoPayIds are deleted successfully</responseMessage>
</C1-DeleteAutoPayService>
```

Sample AutoPay Deletion Response for an AutoPayId in XML Format

```
<C1-DeleteAutoPayService>
    <accountautopayid>
        <AccountAutoPayId>7779143568</AccountAutoPayId>
    </accountautopayid>
    <responseStatus>Success</responseStatus>
    <responseMessage>AutoPayId is deleted successfully</responseMessage>
</C1-DeleteAutoPayService>
```

Note: We have given the above sample to illustrate how the tags should be nested while sending an auto pay deletion request from the external system. However, you should not seamlessly use this sample in your environment because the required pre-requisite data may not be available in your environment. In such case, the system will give erroneous results.

Sample AutoPay Deletion Request and Response in JSON Format

Using the below example, you can delete an autopay request which has incorrect data.

Sample AutoPay Deletion Request for an AccountId in JSON Format

```
{
    -"C1-DeleteAutoPayService":
    {
        -"accountId": -"8889143815",
        -"accountIdentifierType": -"ACCTNUM",
        -"accountNumber": -"2345678000",
        -"division": -"D0301",
        -"sourceSystem": -"DS01",
        -"c1ExternalTransactionId": -"787878",
        -"externalSourceId": -"2240",
    -}
}
```

Sample AutoPay Deletion Request for an AutoPayId in JSON Format

```
{
    -"C1-DeleteAutoPayService":
    {
        -"accountId": -"8889143815",
        -"accountIdentifierType": -"ACCTNUM",
        -"accountNumber": -"2345679000",
        -"division": -"D0401",
        -"sourceSystem": -"ES01",
        -"c1ExternalTransactionId": -"989898",
        -"externalSourceId": -"2250",
        -"accountautopaydata":
        [
            {
                -"accountAutoPayId": -"7779143568"
            -}
        -]
    -}
}
```

Sample AutoPay Deletion Response for an AccountId in JSON Format

```
{
    -"C1-DeleteAutoPayService":
```

Copyright © 2024, Oracle and/or its affiliates.

```
    {
        {
            -"AccountAutoPayId": -"8889143891,12333344554"
        -}
        -"responseStatus": -"Success",
        -"responseMessage": -"AutoPayIds are deleted successfully"
    -}
}
```

Sample AutoPay Deletion Response for an AutoPayId in JSON Format

```
{
    -"C1-DeleteAutoPayService":
    {
        {
            -"AccountAutoPayId": -"7779143568"
        -}
        -"responseStatus": -"Success",
        -"responseMessage": -"AutoPayId is deleted successfully"
    -}
}
```

Note: We have given the above sample to illustrate how the attribute value pairs should be structured while sending an auto pay deletion request from the external system. However, you should not seamlessly use this sample in your environment because the required pre-requisite data may not be available in your environment. In such case, the system will give erroneous results.

C1-ViewAutoPay Business Service

The **Account AutoPay Business Service for View (C1-ViewAutoPayService)** inbound web service is used for viewing entire auto pay details on an account, in the Oracle Revenue Management and Billing application from an external system. You can provide an accountId or account identifier type along with account number and division to the inbound web service. If both, accountId and account identifier type are provided, accountId is given first priority ahead of the account identifier type.

You can view the entire auto pay details on an account, through the following scenarios:

- If you provide either an accountId or account identifier type along with the account number as a request, the business service fetches all of the auto pay details present on the account and returns a list of all the auto pay Ids in the from of response.
- If start date is provided as a request, the business service returns a list of auto pay Ids which were created for a particular account, based on the start date.
- If auto pay source is provided as a request, the business service returns a list of auto pay Ids which were created for a particular account, based on the auto pay source information.
- If both, start date and the auto pay source is provided as a request, the business service returns a list of auto pay Ids which were created for a particular account, based on the start date and auto pay source information.

If the account validation fails, the inbound service returns the following response:

- responseMessage: {error message is returned}

AutoPay View Request in XML Format

The following code snippet illustrates how the tags should be nested while receiving the autopay view request in the XML format:

```
<C1-ViewAutoPayService>
    <accountIdentifierType></accountIdentifierType>
    <accountNumber></accountNumber>
    <division></division>
```

Copyright © 2024, Oracle and/or its affiliates.

```
    <AccountId></AccountId>
    <sourceSystem></sourceSystem>
    <c1ExternalTransactionId></c1ExternalTransactionId>
    <externalSourceId></externalSourceId>
    <autopaySource></autopaySource>
    <startDate></startDate>
 </C1-ViewAutoPayService>
```

AutoPay View Response in XML Format

The following code snippet illustrates how the tags should be nested while sending the autopay view response in the XML format:

```
<C1-ViewAutoPayService>
    <statusCd></statusCd>
    <accountautopaydata>
        <accountId></accountId>
        <accountAutoPayId></accountAutoPayId>
        <startDate></startDate>
        <autopaySource></autopaySource>
        <autopayRouteType></autopayRouteType>
        <percent></percent>
        <priorityNum></priorityNum>
        <expiresOn></expiresOn>
        <externalAccountId></externalAccountId>
        <autopayMaxWithdrawalAmount></autopayMaxWithdrawalAmount>
        <autoPayMethod></autoPayMethod>
        <entityName></entityName>
        <comments></comments>
    </accountautopaydata>
    <accountautopaydata>
        <accountId></accountId>
        <accountAutoPayId></accountAutoPayId>
        <startDate></startDate>
        <endDate></endDate>
        <autopaySource></autopaySource>
        <autopayRouteType></autopayRouteType>
        <percent></percent>
        <priorityNum></priorityNum>
        <expiresOn></expiresOn>
        <externalAccountId></externalAccountId>
        <autopayMaxWithdrawalAmount></autopayMaxWithdrawalAmount>
        <autoPayMethod></autoPayMethod>
        <entityName></entityName>
        <comments></comments>
        <accountautopaymentchar>
            <characteristicType></characteristicType>
            <sequence></sequence>
            <characteristicValue></characteristicValue>
        </accountautopaymentchar>
    </accountautopaydata>
</C1-ViewAutoPayService>
```

AutoPay View Request in JSON Format

The following code snippet illustrates how the tags should be nested while receiving the autopay view request in the JSON format:

```
{
    -"C1-ViewAutoPayService":
    {
        -"accountIdentifierType": -"",
```

Copyright © 2024, Oracle and/or its affiliates.

```
        -"accountNumber": -"",
        -"division": -"",
        -"AccountId": -"",
        -"sourceSystem": -"",
        -"c1ExternalTransactionId": -"",
        -"externalSourceId": -"",
        -"AutoPaySource": -"",
        -"StartDate": -""
    -}
}
```

AutoPay View Response in JSON Format

The following code snippet illustrates how the tags should be nested while sending the autopay view response in the JSON format:

```
{
    -"C1-ViewAutoPayService":
    {
        -"statusCd": -"",
        -"accountautopaydata":
        [
            {
                -"accountId": -"",
                -"accountAutoPayId": -"",
                -"startDate": -"",
                -"autopaySource": -"",
                -"autopayRouteType": -"",
                -"percent": -"",
                -"priorityNum": -"",
                -"expiresOn": -"",
                -"externalAccountId": -"",
                -"autopayMaxWithdrawalAmount": -"",
                -"autoPayMethod": -"",
                -"entityName": -"",
                -"comments": -""
            -},
            {
                -"accountId": -"",
                -"accountAutoPayId": -"",
                -"startDate": -"",
                -"endDate": -"",
                -"autopaySource": -"",
                -"autopayRouteType": -"",
                -"percent": -"",
                -"priorityNum": -"",
                -"expiresOn": -"",
                -"externalAccountId": -"",
                -"autopayMaxWithdrawalAmount": -"",
                -"autoPayMethod": -"",
                -"entityName": -"",
                -"comments": -"",
                -"accountautopaymentchar":
                {
                    -"characteristicType": -"",
                    -"sequence": -"",
                    -"characteristicValue": -""
                -}
            -}
        -]
    -}
}
```

Copyright © 2024, Oracle and/or its affiliates.

AutoPay View Request and Response - Tags in XML Format

Note: We recommend you to refer the AutoPay View Request in XML Format on page 52 and AutoPay View Response in XML Format on page 53 topics in parallel while understanding the below mentioned tags. This will help you to understand how the tags are nested in the XML format.

Before calling the **C1-ViewAutoPayService** business service, you need to ensure that the autopay view request contains the following tags:

Tag Name	Tag Description	Mandatory (Yes or No)
division	Used to indicate the division to which the account belongs. **Note:** You must specify the division code which is already defined in the system.	Yes (Conditional) **Note:** This data is required while adding or updating the autopay details of an account.
accountId	Used to indicate the account for which the autopay details are added or updated. **Note:** You must specify the accountId which is already defined in the system.	Yes (Conditional) **Note:** This data is required if **accountIdentifierType** and **accountNumber** is not specified.
accountIdentifierType	Used to specify the account identifier type. **Note:** You must specify an account identifier type which is already defined in the system.	Yes (Conditional) **Note:** This data is required while deleting the auto pay details of an account.
accountNumber	Used to indicate the account for which you want to create or update the autopay information.	Yes (Conditional) **Note:** This data is required when the **accountIdentifierType** is specified.
StartDate	Used to indicate the date from when the autopay is effective for an account.	No
AutopaySource	Used to specify the autopay source from where the auto pay information is received for an account.	No
sourceSystem	Used to indicate the external system from where the autopay information is received.	No
c1ExternalTransactionId	Used to indicate the transaction in the external source system from where the autopay information is received.	No
externalSourceId	Used to specify the external source system Id.	No
accountautopaydata	Used to specify a list of autopay details for an account.	No
accountId	Used to indicate the accountId	No
accountAutoPayId	Used to specify the autopay Id for an account.	No

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Tag Description	Mandatory (Yes or No)
	Note: The accountAutoPayId is provided as an input parameter whenever the autopay details are either updated or deleted for an account.	
startDate	Used to indicate the date from when the autopay is effective for an account. **Note:** The startDate cannot not be later than the endDate.	No
endDate	Used to indicate the date till when the autopay is effective for an account. **Note:** The startDate cannot not be earlier than the endDate.	No
autopaySource	Used to indicate the source of funds used to make the automatic payment. For example, debit card or credit card. **Note:** You must specify the autopaySource which is already defined in the system.	No
percent	Used to indicate the percentage in which the payment must be split between two or more autopay sources having the same priority.	No
priorityNum	Used to indicate the priority in which the autopay request should be considered when multiple auto pay requests are effective on the bill's due date. **Note:** You must specify the priorityNum which is already defined in the system without the occurrence of a decimal.	Yes
expiresOn	Used to specify the date when the autopay source will expire.	No
externalAccountId	Used to indicate the account in the external system for which the auto pay details should be added or updated.	Yes
autopayMaxWithdrawalAmount	Used to specify the maximum amount that can be automatically debited from the autopay account. It is used to set the limit on the withdrawal amount. **Note:** The autopay max withdrawal amount must be a positive number with or without a decimal. It should not be less than zero.	No
autoPayMethod	Used to indicate how you want to process the autopay request. The valid values are: • **C1DD** - Indicates that you want to directly debit the main customer's bank account. • **C1PA** - Indicates that you want to send a payment advice to the main customer.	No

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Tag Description	Mandatory (Yes or No)
	Note: You must specify the autopay method which is already defined in the system.	
entityName	Used to specify the name of the person.	No
comments	Used to specify additional information about the autopay instruction.	No
accountautopaymentchar	Used to specify a list of autopay characteristics for an account.	No
accountAutoPayId2	Used to specify the autopayId for an account. **Note:** The accountAutoPayId is provided as an input parameter whenever the auto pay details are either updated or deleted for an account.	No
characteristicType	You must specify a characteristic type where the characteristic entity is set to **Account**. **Note:** You must specify the characteristic type which is already defined in the system.	No
sequence	Used to indicate the sequence in which the autopay requests must be executed.	No
characteristicValue	Used to specify the value for the characteristic type.	No

The following table lists and describes the tags which appear in the autopay view response in the XML format:

Tag Name	Tag Description
accountautopayid	Displays the accountautopayid of the account.
response	Displays the complete autopay details of the account.
statusCd	Indicates whether the autopay viewing request is successfully processed or not. The valid values are: • Success • Fail
messageText	Indicates an error that occurred while viewing the autopayIds. **Note:** This tag appears when the value in the **responseStatus** tag is set to **Fail**.

Related Topics

For more information on...	See...
Sample AutoPay View Request and Response in XML Format	Sample AutoPay Add/Update Request and Response in XML Format on page 40

Copyright © 2024, Oracle and/or its affiliates.

AutoPay View Request and Response - Attribute/Value Pairs in JSON Format

Note: We recommend you to refer the AutoPay View Request in JSON Format on page 53 and AutoPay View Response in JSON Format on page 54 topics in parallel while understanding the below mentioned attributes. This will help you to understand how the attribute/value pairs are structured in the JSON format.

Before calling the **C1-ViewAutoPayService** business service, you need to ensure that the autopay view request contains the following attributes:

Attribute Name	Attribute Description	Mandatory (Yes or No)
schema	Used to specify the tags of the cancel adjustment inbound web service.	Yes
division	Used to indicate the division to which the account belongs.	Yes (Conditional)
	Note: You must specify the division code which is already defined in the system.	**Note:** This data is required while adding or updating the auto pay details of an account.
accountId	Used to indicate the account for which the auto pay details are added or updated.	Yes (Conditional)
	Note: You must specify the accountId which is already defined in the system.	**Note:** This data is required if **accountIdentifierType** and **accountNumber** is not specified.
accountIdentifierType	Used to specify the account identifier type.	Yes (Conditional)
	Note: You must specify an account identifier type which is already defined in the system.	**Note:** This data is required while deleting the auto pay details of an account.
accountNumber	Used to indicate the account for which you want to create or update the auto pay information.	Yes (Conditional)
		Note: This data is required when the **accountIdentifierType** is specified.
StartDate	Used to indicate the date from when the auto pay is effective for an account.	No
AutopaySource	Used to specify the autopay source from where the auto pay information is received for an account.	No
sourceSystem	Used to indicate the external system from where the auto pay information is received.	No
c1ExternalTransactionId	Used to indicate the transaction in the external source system from where the auto pay information is received.	No
externalSourceId	Used to specify the external source system Id.	No
accountautopaydata	Used to specify a list of auto pay details for an account.	No
accountId	Used to indicate the accountId	No
accountAutoPayId	Used to specify the auto pay Id for an account.	No

Copyright © 2024, Oracle and/or its affiliates.

Attribute Name	Attribute Description	Mandatory (Yes or No)
	Note: The accountAutoPayId is provided as an input parameter whenever the auto pay details are either updated or deleted for an account.	
startDate	Used to indicate the date from when the auto pay is effective for an account. **Note:** The startDate cannot not be later than the endDate.	No
endDate	Used to indicate the date till when the auto pay is effective for an account. **Note:** The startDate cannot not be earlier than the endDate.	No
autopaySource	Used to indicate the source of funds used to make the automatic payment. For example, debit card or credit card. **Note:** You must specify the autopaySource which is already defined in the system.	No
percent	Used to indicate the percentage in which the payment must be split between two or more auto pay sources having the same priority.	No
priorityNum	Used to indicate the priority in which the auto pay request should be considered when multiple auto pay requests are effective on the bill's due date. **Note:** You must specify the priorityNum which is already defined in the system without the occurrence of a decimal.	Yes
expiresOn	Used to specify the date when the auto pay source will expire.	No
externalAccountId	Used to indicate the account in the external system for which the auto pay details should be added or updated.	Yes
autopayMaxWithdrawalAmount	Used to specify the maximum amount that can be automatically debited from the auto pay account. It is used to set the limit on the withdrawal amount. **Note:** The auto pay max withdrawal amount must be a positive number with or without a decimal. It should not be less than zero.	No
autoPayMethod	Used to indicate how you want to process the automatic payment request. The valid values are: • **C1DD** - Indicates that you want to directly debit the main customer's bank account. • **C1PA** - Indicates that you want to send a payment advice to the main customer. **Note:** You must specify the auto pay method which is already defined in the system.	No

Copyright © 2024, Oracle and/or its affiliates.

Attribute Name	Attribute Description	Mandatory (Yes or No)
entityName	Used to specify the name of the person.	No
comments	Used to specify additional information about the auto pay instruction.	No
accountautopaymentchar	Used to specify a list of auto payment characteristics for an account.	No
accountAutoPayId2	Used to specify the auto pay Id for an account. **Note:** The accountAutoPayId is provided as an input parameter whenever the auto pay details are either updated or deleted for an account.	No
characteristicType	You must specify a characteristic type where the characteristic entity is set to **Account**. **Note:** You must specify the characteristic type which is already defined in the system.	No
sequence	Used to indicate the sequence in which the auto pay requests must be executed.	No
characteristicValue	Used to specify the value for the characteristic type.	No

The following table lists and describes the attributes which appear in the autopay view response in the JSON format:

Attribute Name	Attribute Description
accountautopayid	Displays the accountautopayid of the account.
response	Displays the complete autopay details of the account.
statusCd	Indicates the status returned by the business service. The valid values are: • Success • Fail
messageText	Indicates an error that occurred while viewing the autopayIds. **Note:** This attribute appears when the value in the **responseStatus** attribute is set to **Fail**.

Related Topics

For more information on...	See...
Sample AutoPay View Request and Response in JSON Format	Sample AutoPay View Request and Response in JSON Format on page 61

Sample AutoPay View Request and Response in XML Format

Using the below example, you can view the autopay information for the 5264526467 account.

Sample AutoPay View Request in XML Format

```
<C1-ViewAutoPayService>
    <accountIdentifierType>ACCTNUM</accountIdentifierType>
    <accountNumber>2345678000</accountNumber>
    <division>D0301</division>
```

Copyright © 2024, Oracle and/or its affiliates.

```
    <AccountId>5264526467</AccountId>
    <sourceSystem>DS01</sourceSystem>
    <c1ExternalTransactionId>787878</c1ExternalTransactionId>
    <externalSourceId>2240</externalSourceId>
    <autopaySource>CITIBANKCHK</autopaySource>
    <startDate>2010-09-06</startDate>
 </C1-ViewAutoPayService>
```

Sample AutoPay View Response in XML Format

```
<C1-ViewAutoPayService>
    <statusCd>Success</statusCd>
    <accountautopaydata>
        <accountId>5264526467</accountId>
        <accountAutoPayId>5966771784</accountAutoPayId>
        <startDate>2020-01-01</startDate>
        <autopaySource>Citibank, NA</autopaySource>
        <autopayRouteType>Automated Clearing House</autopayRouteType>
        <percent>100.00</percent>
        <priorityNum>1</priorityNum>
        <expiresOn>11-2030</expiresOn>
        <externalAccountId>77887765a2231</externalAccountId>
        <autopayMaxWithdrawalAmount>0.00</autopayMaxWithdrawalAmount>
        <autoPayMethod>Direct Debit</autoPayMethod>
        <entityName>xUpload Inc 250</entityName>
        <comments>Test</comments>
    </accountautopaydata>
    <accountautopaydata>
        <accountId>5264526467</accountId>
        <accountAutoPayId>5966771975</accountAutoPayId>
        <startDate>2020-01-01</startDate>
        <endDate>2020-03-31</endDate>
        <autopaySource>Citibank, NA</autopaySource>
        <autopayRouteType>Automated Clearing House</autopayRouteType>
        <percent>100.00</percent>
        <priorityNum>10</priorityNum>
        <expiresOn>11-2025</expiresOn>
        <externalAccountId>123456</externalAccountId>
        <autopayMaxWithdrawalAmount>12030.00</autopayMaxWithdrawalAmount>
        <autoPayMethod>Direct Debit</autoPayMethod>
        <entityName>Autopay Split, VQ</entityName>
        <comments>Test</comments>
        <accountautopaymentchar>
            <characteristicType>SIFS -- Suffix Value</characteristicType>
            <sequence>1</sequence>
            <characteristicValue>MN</characteristicValue>
        </accountautopaymentchar>
    </accountautopaydata>
</C1-ViewAutoPayService>
```

Note: We have given the above sample to illustrate how the tags should be nested while sending an auto pay view request from the external system. However, you should not seamlessly use this sample in your environment because the required pre-requisite data may not be available in your environment. In such case, the system will give erroneous results.

Sample AutoPay View Request and Response in JSON Format

Using the below example, you can view the autopay information for the 5264526467 account.

Sample AutoPay View Request in JSON Format

```
{
    -"C1-ViewAutoPayService":
```

Copyright © 2024, Oracle and/or its affiliates.

```
    {
        -"accountIdentifierType": -"ACCTNUM",
        -"accountNumber": -"2345678000",
        -"division": -"D0301",
        -"AccountId": -"5264526467",
        -"sourceSystem": -"DS01",
        -"c1ExternalTransactionId": -"787878",
        -"externalSourceId": -"2240",
        -"AutoPaySource": -"CITIBANKCHK",
        -"StartDate": -"2010-09-06"
    -}
}
```


Sample AutoPay View Response in JSON Format


```
{
    -"C1-ViewAutoPayService":
    {
        -"statusCd": -"Success",
        -"accountautopaydata":
        [
            {
                -"accountId": -"5264526467",
                -"accountAutoPayId": -"5966771784",
                -"startDate": -"2020-01-01",
                -"autopaySource": -"CitiBank, NA",
                -"autopayRouteType": -"Automated Clearing House",
                -"percent": -"100.00",
                -"priorityNum": -"1",
                -"expiresOn": -"11-2030",
                -"externalAccountId": -"77887765a2231",
                -"autopayMaxWithdrawalAmount": -"0.00",
                -"autoPayMethod": -"Direct Debit",
                -"entityName": -"xUpload Inc 250",
                -"comments": -"Test"
            -},
            {
                -"accountId": -"5264526467",
                -"accountAutoPayId": -"5966771975",
                -"startDate": -"2020-01-01",
                -"endDate": -"2020-03-31",
                -"autopaySource": -"Citibank, NA",
                -"autopayRouteType": -"Automated Clearing House",
                -"percent": -"100.00",
                -"priorityNum": -"10",
                -"expiresOn": -"11-2025",
                -"externalAccountId": -"123456",
                -"autopayMaxWithdrawalAmount": -"12030.00",
                -"autoPayMethod": -"Direct Debit",
                -"entityName": -"Autopay Split, VQ",
                -"comments": -"Test Comments",
                -"accountautopaymentchar":
                {
                    -"characteristicType": -"SIFS -- Suffix Value",
                    -"sequence": -"1",
                    -"characteristicValue": -"MN"
                -}
            -}
        -]
    -}
}
```


Copyright © 2024, Oracle and/or its affiliates.

> **Note:** We have given the above sample to illustrate how the attribute value pairs should be structured while sending an auto pay view request from the external system. However, you should not seamlessly use this sample in your environment because the required pre-requisite data may not be available in your environment. In such case, the system will give erroneous results.

Using The Account / Person Replicator

There are two different ways to use this transaction:

- You can make many copies of a given Person / Account combination. This may be required if your company works with 3 rd party agencies who enroll customers on its behalf. In this situation, the 3 rd party agency may receive blocks of pre-allocated account numbers to be used when they enroll a customer. When a customer signs up, the person information can be updated with information about the newly enrolled customer.
- Perform the following steps if you need to create many copies of a given person and account:
  - Add a new person and account if you don't already have a customer to serve as your "template".
  - Use the account / person replicator to create copies of the person and the account as described below (and make sure to use the `Replicate Acct and Person` option).
- You can create many Accounts for a single Person. This may be required if you have a customer who wants many separate accounts rather than a single consolidated account.
- Perform the following steps if you need to create many copies of a given account for a specific person:
  - Select the account that serves as the template account for the person. Note, this account must be linked to the person who wants the many accounts.
  - Use the account / person replicator to create copies of the account as described below (and make sure to use the `Replicate Account Only` option).

Use **Customer Management, Account/Person Replicator** to open this page.

Description of Page

Choose the **Account** that serves as the template account.

> **Note: Recommendation.** Carefully verify that the template account you have chosen is correct before you save the replicated accounts. After you save the replicated accounts, any corrections could prove time consuming.

At the top of the page, indicate the **Number** of accounts and persons **to Replicate**.

Indicate the **Main Person Name Base**, which will be used as the main person name for all the new persons to be created. This field will be protected if you choose a **Replicate Type** of `Replicate Account Only` (as no persons are created).

Use **Replicate Type** to define if you want to create multiple copies of an account for a given person or if you want to create many new person / account combinations. See the description that appears above for more information about this field.

After entering the above, click the **Replicate** button to generate your new accounts and persons. The grid displays informational messages describing what will happen when you click save. For example, it will verify how many accounts will be created, the person whose record will be copied, the name that will be used on all the new persons, etc..

If everything looks clean, click save.

Information Replicated for New Persons

Except for the main name, which you indicate in the **Main Person Name** field, all information for the main person linked to the account will be copied to the new persons.

Copyright © 2024, Oracle and/or its affiliates.

CAUTION: This copy includes the person's identification number. This will result in multiple persons with the same identification number. You should ensure that the identification number for your template person is a temporary number. When you sign up your actual customer, be sure to update the identification number to a valid number for the person.

Information Replicated for New Accounts

Although most information for an account will be copied, not all information will be. The following points describe the information that is NOT copied on newly replicated accounts:

- Mailing location will be set to blank
- Setup Date is set to the current date
- Credit Review Date and Postpone Credit Review Until Date will be set to blank
- Auto Pay information will not be copied
- Credit Rating History will not be copied
- Non-cash deposit information will not be copied

Note: **Note** that all miscellaneous persons linked to the account (i.e., those who are not flagged as Main Customers) will be linked to the new accounts. With this functionality, for example, you can create a person for a third party enroller and link this enroller to the template account. This enroller will then be linked to all the new accounts.

Maintaining Contracts

A contract contains the terms and conditions controlling how the system calculates charges for a specific service supplied to a customer. The topics in this section describe the pages on which contract related information is maintained.

Note: **The system creates most contracts behind the scenes.** Most contracts are created "behind the scenes" by the system as part of the Start Service and Order completion processes. You should only have to access the contract pages if you need to fine-tune what the system defaults. Refer to Start / Stop and Maintaining Orders for the details.

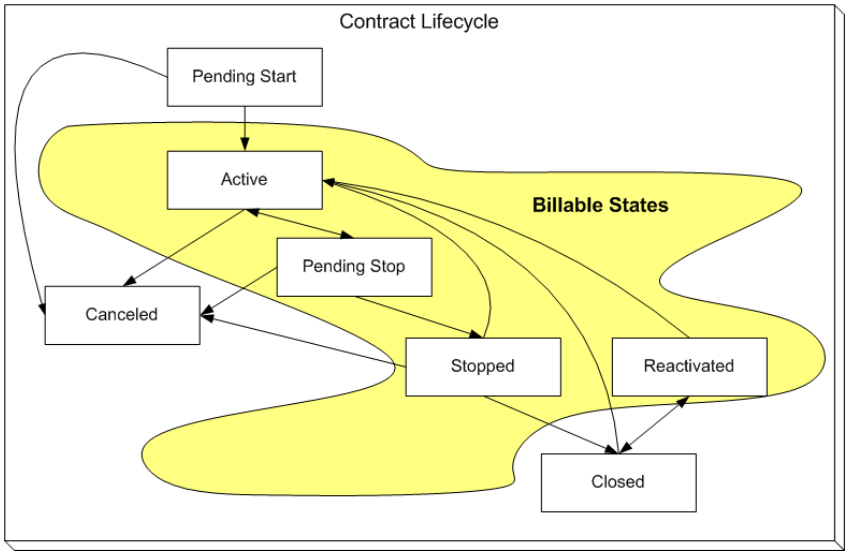
Fastpath: For more information about contracts and their place in the "V", refer to Understanding The "V".

CAUTION: Take special care when adding a new contract to specify the appropriate Division and Contract Type as it will affect how the customer is billed, what rate can be specified, how overdue debt is collected, and much more. After the contract is activated, you may not change its contract type.

The Lifecycle Of A Contract

The following diagram shows the possible lifecycle of a contract.

Copyright © 2024, Oracle and/or its affiliates.



`Pending Start` When you initiate a request to start service, the system initiates a contract in the state of `Pending Start`. A pending start contract remains in this state until everything necessary to start service is defined in the system.

`Active` The system automatically changes the state of `Pending Start` contracts to `Active` when everything necessary to start service exists in the system. Refer to Activating Pending Starts for more information. While in this state, the system bills the contract's account for the service.

`Pending Stop` When you initiate a request to stop service, the system progresses the contract to the `Pending Stop` state. A pending stop contract remains in this state until everything necessary to stop service is defined in the system.

`Stopped` The system automatically changes the state of `Pending Stop` contracts to `Stopped` when everything necessary to stop service exists in the system. Refer to Finalizing Pending Stops for more information. The contract remains in this state until it is paid in full.

`Closed` The system moves a contract to the `Closed` state when it no longer has a financial balance.

`Reactivated` The system moves a contract to the `Reactivated` state if a financial transaction is created after a contract is `Closed` (e.g., when a payment is canceled after the contract is closed). Please be aware that `Reactivated` contracts are processed by the Write Off Monitor.

When the contract once again has a zero balance (e.g., because the customer pays or the debt is written off), the contract returns to the `Closed` state.

`Cancelled` You may `Cancel` a contract if it shouldn't exist (because the customer doesn't want service). While cancellation doesn't physically remove a contract from the database, most queries and trees suppress information about canceled contracts. This means cancellation is a nice way to logically delete undesired contracts.

You'll notice in the above diagram that you can cancel an `Active, Pending Stop` or `Stopped` contract. You may only do this if the contract's financial transactions and non-billed billable charges are all cancelled.

You should rarely have to change a contract's status. Why? Because the system changes the status behind-the-scenes when various events take place. For example,

- When you request a service start using the Start/Stop page, the system creates a contract in the `pending start` state.
- If you cancel a pending start before service is activated, the system automatically changes the state of the contract to `cancelled`.
- When everything necessary to start service exists, the system automatically changes the state of the contract to `active`.
- When you request a service stop using the Start/Stop page, the system changes the state of the stopped contract to `pending stop` state.
- When everything necessary to stop service exists, the system automatically changes the state of the contract to `stopped`.

Copyright © 2024, Oracle and/or its affiliates.

- When a final payment is received and the stopped contract's balance becomes zero, the system automatically changes the state of the contract to `closed`. The system can also `close` a contract as part of the write-off processing.
- If the payment bounces, the system automatically changes the state of the contract to `reactivated`. And the system will change it back to `closed` when the account's balance returns to zero.
- If a billable charge is interfaced into the system, the system automatically changes the state of the contract to `reactivated`.

You may need to manually change a contract's state for the following reasons:

- If a contract was activated inadvertently, you may cancel it (and therefore logically remove it from the system) after canceling all of its financial transactions. You do this by clicking the cancel button on the Contract page. You may also cancel an incorrect contract if the status transitions to pending stop or stopped before you have a chance to correct the mistake.
- If you don't want to wait for the system to automatically activate a pending start contract, you may push the activate button on the contract page. You might do this in order to create a bill for an impatient customer.
- If you don't want to wait for the system to automatically finalize a pending stop contract, you may push the stop button on the contract page. You might do this in order to create a real time final bill for a customer.
- If you don't want to wait for the system to automatically close a stopped contract, you may push the close button on the contract page.
- Sometimes a contract is closed or `stopped` by mistake. You can activate a stopped, closed, or reactivated contract using the **Reinstate Contract** button on the contract page. Using the **Reinstate Contract** button retains the current service's billing history.

Note: One time invoices (i.e., one-time billable charges) are interesting. One-time invoice contracts follow the above lifecycle; it's just that they do it quickly. You see, when a one-time invoice contract is created, both its start and stop dates are known (they are the invoice date). If you followed the above state transition discussion, you'll understand that the system progresses these invoices from pending start to stopped when the contract is activated.

Renewing Contracts

Some contract types (e.g., non-billed budgets) may allow or require renewal. If renewal is optional, a renewal date may be specified when the contract is created. If renewal is required, you must specify a renewal date.

The *Contract renewal background process* initiates the renewal for all `active` contracts when the renewal date is reached (i.e., the renewal date is on or before the process date). When a contract is renewed:

- The contract renewal algorithm (specified on the contract type) is executed. The contract renewal algorithm can be used to customize the processing required to renew an contract.
- If the contract renewal algorithm is successful, the renewal and expiration date (if calculated by the renewal algorithm) on the contract are updated with the values returned from the renewal algorithm.
- If the renewal process is not successful, a To Do list entry (of type TD-SARN) is created for the account and contract.

Expiring Contracts

Contracts may specify an expiration date. Some contract types (e.g., non-billed budgets) require an expiration date. The *Contract expiration background process* initiates the request to stop service for all `active` Contracts when the expiration date is reached (i.e. the expiration date is on or before the process date).

Contract - Main Information

The Main page contains basic contract information. Open **Customer Management, Contract** to maintain this information.

Description of Page

Contract (contract) **Info** and **Contract ID** are displayed on every page. These values only appear after the contract exists on the database. The **ID** is a system assigned random number that stays with a contract for life. The **Contract Info** is a concatenation of important details about the contract and its account.

Copyright © 2024, Oracle and/or its affiliates.

Note: Formatting may be performed by a plug-in. The contents of **Contract Info** may be formatted by a plug-in algorithm on contract type. Refer to the base package's C1-SAT-INFO for an example. If such an algorithm is not plugged-in on the contract type, the system looks for a corresponding algorithm on the installation record. Refer to the base package's C1-SAI-INFO for an example. If you prefer different formatting logic, your system administrator should configure the system appropriately.

Contract Status defines the state of the contract. The system typically changes the state of a contract behind-the-scenes. If you can't wait for the system, use one of the adjacent buttons:

- Click the **Activate Contract** button to activate a `Pending Start` contract.
- Click the **Cancel Contract** button to cancel a `Pending Start` or `Active`, `Pending Stop` or `Stopped` contract.
- Click the **Stop Contract** button to stop a `Pending Stop` contract.
- Click the **Close Contract** button to close a `Stopped` or `Reactivated` contract.
- Click the **Reinstate Contract** button to reinstate a `Closed`, `Reactivated`, or `Stopped` contract to an `Active` contract.

Fastpath: Refer to The Lifecycle of a Contract for more information about **Contract Status**.

If the quotation module is not turned off, a "real" contract can be created when a *proposal contract* is accepted. If the contract you've displayed was created in such a way, information about the **Proposal Contract** will appear beneath **Contract Status**.

Enter the **Account ID** that is financially responsible for the contract. If you change a contract's **Account ID**, you are effectively transferring this contract (and its debt) to the new account.

The **Policy ID** field indicates the policy for which the contract is created.

Indicate the **Division** and **Contract Type**. These fields are very important as they control numerous aspects of the contract's behavior. These fields are gray when the status is not `Pending Start`.

Fastpath: For more information about what the contract type controls, refer to Setting Up Contract Types.

The **Start Date** defines when the financial relationship begins. The **End Date** defines when the financial relationship terminates. If the end date is blank, the contract has not yet been terminated.

Use **Maximum Bill Threshold** if you want the system to generate a bill error when a bill segment is produced *in batch* that exceeds a given value. These bill errors will appear on the standard billing queries and To Do lists. If, after reviewing the high value bill segment, a user truly intends to send the bill out, they should regenerate the bill. Refer to How To Correct A Bill Segment That's In Error for more information.

Note: The value of **Maximum Bill Threshold** defaults from the contract's contract type.

If the **Contract Type** has a special role of `Interval`, you must enter the **Cutoff Time** and **Start Day Option** that is used by interval pricing and derivation algorithms to determine the start and end intervals to process.

- The **Cutoff Time** is entered in "legal time". This value defaults from the installation record.
- For **Start Day Option**, you may choose `Current Day` or `Previous Day`. This value defaults from the installation record.

The payment upload process uses this field to locate the accounts for payments that reference a legacy account number.

If the contract requires a total amount to bill, the amount is displayed. This field is typically only used on loan or cash deposit contracts. It holds either the loan amount or the desired cash deposit. If you increase this field's value, the customer will be billed for any difference between the new amount and the total amount billed (and paid for) to date. The label that appears for this field is defined on the contract type (on the Billing tab). We recommend that the system be set up so that this label is very specific, for example:

- It should be called **Cash Deposit Amount** if the total amount to bill is for a deposit.
- It should be called **Loan Amount** if the total amount to bill is for a loan.

Copyright © 2024, Oracle and/or its affiliates.

Number of Payment Periods only appears for loan contract types. It displays the number of periods specified when the loan was first activated. (The number of payment periods may have been subsequently changed on Loan maintenance.)

If the contract uses a recurring charge, the current recurring charge amount is displayed. Transfer to the Contract - Recurring Charge to update this value. This field is not visible if the contract's contract type doesn't allow recurring charges. The label that appears for this field is defined on the contract type (on the Billing tab). We recommend that the system be set up so that this label is very specific, for example:

- It should contain **Budget Amount** if the recurring charge is for a budget.
- It should contain **Installment Amount** if the recurring charge is used for a deposit being paid in installments.
- It should contain **Payment Amount** if the recurring charge is used for a loan.

CIAC Review Date displays the last date that the Contributions In Aid of Construction (CIAC) review process ran for this contract. This field is only visible if the contract type indicates this is a CIAC contract.

Deposit Interest Calculated contains the last date the system applied interest. This field is only visible if the contract type has a special role of `Cash Deposit`.

The **Sched** uled **Payment Autopay** flag determines if the non-billed budget scheduled payments are excluded from automatic payment or included by automatic payment. If the account is not set up for automatic payment for the period that covers the non-billed budget, this flag cannot be set. This field is only visible if the contract has a special role of `Non-billed Budget`.

The **NBB Recommendation Rule** displays the description of the non-billed budget recommendation rule used for this contract. This field is only visible if the contract has a special role of `Non-billed Budget`.

Expiration Date contains the date at which the contract is set to expire.

Renewal Date contains the date at which the contract is to be renewed. Renewal date is disabled if renewal is not allowed on the contract type.

Debt Class indicates the debt class into which the contract's current balance is categorized by credit and collections. A contract's debt class is defined on its contract type. Refer to Different Collection Criteria For Different Customers And Different Debt for more information about debt class.

Current Balance indicates the amount the customer currently owes for the service.

Payoff Balance indicates the amount the customer would owe if they wanted to close the contract. This value is only displayed when it differs from current balance.

Fastpath: Refer to Current Amount versus Payoff Amount for more information.

The **Days Old** and **Arrears Amount** information shows how old the current balance is. This information will be blank if the current balance is a credit.

Note: Information about debt balances by age is not applicable to open-item accounts.

This bottom of this page contains a tree that shows the various objects linked to the contract. You can use this tree to both view high-level information about these objects and to transfer to the respective page in which an object is maintained.

Points to Note:

There are certain division-specific validations that are triggered on the **Contract** screen. These validations are triggered only when the **Enforce division-specific validations** option type for the **C1_DIVFUNCT** feature configuration is set to **Y**.

The following division-specific validations are triggered on the **Contract** screen:

- The division to which the contract belongs should be the same as the division to which the account belongs.

The **Bill Message** in the **Misc** tab that you select should be valid for the division assigned to the contract.

The topics that follow describe miscellaneous features on this page.

Copyright © 2024, Oracle and/or its affiliates.

The Impact of Proposal Contracts

If the quotation module is not turned off, you may display a *proposal contract* on Contract - Main. This section describes how Contract - Main differs when a proposal contract is displayed.

The **Contract Status** section is replaced with a section that shows the status of the proposal contract (proposal contract).

Proposal Contract Status defines the state of the proposal contract. The Quote Maintenance transaction typically changes the status of a proposal contract when users accept, decline and cancel quote details. If you wish to change the status of a proposal contract independent of this transaction, you can press one of the adjacent buttons

- Click the **Activate** button to change the status from `Pending` to `Quotable`. Note, you can set up the system to automatically make newly created proposal contracts quotable by plugging in the appropriate `Proposal Contract Creation` algorithm on the contract's contract type.
- Click the **Accept** button to change the status from `Quotable` to `Accepted`. When you accept a proposal contract, you will be asked to supply the **Start Date** of the real contract that will be created.
- Click the **Decline** button to change the status from `Quotable` to `Declined`. When you decline a proposal contract, you will be asked to supply a **Decline Reason**.
- Click the **Cancel** button to change the status from `Quotable` to `Canceled`.
- Click the **Reactivate** button to return a `Canceled` or `Declined` proposal contract to the `Quotable` state.

Fastpath: Refer to *Proposal Contract State Transition* for more information about **Proposal Contract Status**.

If the proposal contract has been declined, the **Decline Reason** appears beneath **Proposal Contract Status**. Refer to Accepting / Declining Quote Details for more information.

It the proposal contract was changed after a quote detail was generated using the proposal contract's terms, a **Remark** appears beneath **Proposal Contract Status** highlighting such.

If a real contract was created when the proposal contract was accepted, **Real Contract ID** appears and contains the ID of the real contract. Refer to Accepting / Declining Quote Details for more information.

Contract - Rate Info

The rate info page contains the contract's rate and information needed by the rate to calculate the contract's bill segments. This information is only displayed if the contract's contract type allows a rate.

Open **Customer Management, Contract** and navigate to the **Rate Info** tab to access this page.

Description of Page

The **Rate** grid controls the rate used to calculate the contract's bill segments. Multiple rows will only exist if the contract's **Rate** changes over time.

Billing selects a single rate from the **Rate** grid when it calculates a bill segment for the contract. If multiple rates are defined in this grid, the rate is selected based on the contract's contract type's Rate Selection Date. The following fields display:

- **Effective Date** is the date the rate becomes effective. The contract's start date defaults.
- **Rate Schedule** defines the rate used to calculate the contract's bill segments. Note, you can only choose rates defined as permissible on the contract's contract type; the permissible rates are shown in the adjacent tree.

Note: Possible override. If your contract is linked to a terms of service record, it is possible that the rate used by the system could get overridden by one referenced on the terms of service. If the rate is overridden by the TOS, the tree on this page indicates the rate source.

The **Contract Rider** grid allows you to define the contract riders that are in effect for the contract.

Billing uses this information when it calculates a bill segment for the contract if:

- The rate used to calculate the bill segment contains a rate component that references a bill factor.
- The referenced bill factor requires a contract rider to levy the associated charge.

Copyright © 2024, Oracle and/or its affiliates.

If a **Contract Rider** does not exist or is not effective, the charge associated with the rate component is not applied to the customer's bill.

To modify a contract rider linked to the contract, simply move to a field and change its value. To remove a contract rider, click the - button. To add a new contract rider, click the + button and fill in the information for each field. The following fields display:

- Use **Start Date** to define the date on which the contract rider becomes effective. The contract's start date defaults.
- Use **End Date** to define the date on which the contract rider expires. This field need only be specified if the contract rider expires on an explicit date.
- Use **Bill Factor** to define the type of rider. You may only reference **Bill Factors** designated as being applicable for contract riders. Note, the **Bill Factors** linked to the rate that are marked with contract rider applicability are shown in the adjacent tree under the Contract Rider node.

Fastpath: For more information about contract riders, refer to *Defining General Bill Factor Information*.

The **Contract Values** grid allows you to defined the contract values that are in effect for the contract.

Billing uses this information when it calculates a bill segment for the contract if:

- The rate used to calculate the bill segment contains a rate component that references a bill factor.
- The referenced bill factor uses a contract value to define the amount charged.

The following fields display:

- Use **Start Date** to define the date on which the contract value becomes effective. The contract's start date defaults.
- Use **End Date** to define the date on which the contract value expires. This field need only be specified if the value rider expires on an explicit date.
- Use **Bill Factor** to define the type of value. You may only reference **Bill Factors** designated as allowing values in contract terms. Note, the **Bill Factors** linked to the rate that are marked as allowing values in contract terms are shown in the adjacent tree under the Contract Value node.
- Use **Value** to define contract value.

Fastpath: For more information about contract values, refer to *Defining General Bill Factor Information*.

The **Tax Exemptions** section contains the taxes from which the contract is all or partially exempt.

Billing uses this information when it calculates a bill segment for the contract if:

- The rate used to calculate the bill segment contains a rate component that references a bill factor.
- The referenced bill factor can have tax exemptions that limit the amount of the charge.

If the bill factor doesn't have a tax exemption or there are none that are effective during the bill period, the tax is levied in full. If the contract is exempt, the appropriate charge associated with the tax is calculated. We strongly recommend using the "%T" substitution character in the *Description on Bill* on tax-exemptible rate components so that an appropriate tax exemption message appears on the customer's bill.

Fastpath: For more information about tax exemptions, refer to *Defining General Bill Factor Information*.

The following fields display:

- Use **Start Date** to define the date on which the tax exemption becomes effective.
- Use **End Date** to define the date on which the tax exemption expires.
- Use **Bill Factor** to define the tax from which the contract is exempt. You may only reference bill factors designated as being tax exemptible. Note, the **Bill Factors** linked to the rate that are marked as tax exemptible are shown in the adjacent tree under the Tax Exemption node.
- Use **Percent Exempt** to define the customer tax exemption percent (e.g., 90 means the customer is 90% exempt and will only pay 10% of the normal tax rate).
- Use **Tax Exempt Type** to specify the type of exemption.
- Use **Tax Certification** if the customer has a tax exemption certificate.

Copyright © 2024, Oracle and/or its affiliates.

Rate Information Tree

The tree shows the contract's permissible rates and the related types of contract riders, contract values and tax exemptions that are possible given these rates.

- The Permissible Rates node displays all the rate schedules associated with the contract's contract type.
- The Effective Rate Schedule node indicates the rate schedule currently effective.
- The Contract Rider node displays the bill factors linked to the contract's rate(s) that may have contract riders defined for them. If you expand a node, you can see the individual rate component(s) on which the bill factor is specified. If the contract has a value defined for the contract rider that is currently in effect, it is also shown.
- The Contract Value node displays all the bill factors linked to the contract's rate(s) that may have contract values defined for them. If you expand a node, you can see the individual rate component(s) on which the bill factor is specified. If the contract has a contract value defined that is currently in effect, it is also shown.
- The Tax Exemption node displays all the bill factors linked to the contract's rate(s) that may have tax exemptions defined for them. If you expand a node, you can see the individual rate component(s) on which the bill factor is specified. If the contract has a tax exemption defined for the bill factor that is currently in effect, it is also shown.

Contract - Billing Scenarios

Note: This tab may not appear. This tab is suppressed if the quotation `Contract Management` module is turned off.

This page allows you to define a *proposal contract's* billing scenarios. Refer to *Proposal Contracts Contain Billing Scenarios* for more information.

Open this page using **Customer Management, Contract, Billing Scenarios**.

Description of Page

This page is protected if the contract being displayed is not a *proposal contract*.

When a quote is generated for a proposal contract, the system creates a simulated bill segment for each billing scenario linked to the proposal contract. Each simulated bill segment shows the prospect how much their bill would be if the pricing terms on the proposal contract were applied to the billing scenario's consumption in the related time period.

Note: You should not have to enter this information from scratch. You can set up the system to automatically generate billing scenarios when a proposal contract is created. Refer to *Proposal Contracts Contain Billing Scenarios* for the details.

The following information is defined for each billing scenario:

- **Start Date** and **End Date** define the simulated bill segment's period.
- The grid contains the consumption that will be priced during this period:
  - **Unit of Measure** is the unit of measure of the service quantity.
  - **TOU** is the time-of-use code of the service quantity.
  - **SQI** is the service quantity identifier of the service quantity.
  - **Initial Service Quantity** is the initial quantity passed to the rate (and massaged by the rate's SQ rules).

Contract - Chars, Qty & Rec. Charges

This page contains the Characteristics, Contract Quantities and Recurring Charges that are in effect for the contract.

Use **Customer Management, Contract** and navigate to the **Chars, Qty & Rec. Charges** tab to open this page.

Description of Page

The **Characteristics** grid allows you to define a characteristic for the contract.

Note: You can only choose characteristic types defined as permissible on the contract record. Refer to Setting Up Characteristic Types & Their Values for more information.

Copyright © 2024, Oracle and/or its affiliates.

The following fields display:

Effective Date Indicate the date on which the characteristic value becomes effective. The contract's start date defaults.

Characteristic Type Indicate the type of characteristic.

Characteristic Value Indicate the value of the characteristic.

The **Contract Quantity** grid allows you to define the contract quantities that are in effect for the contract.

Billing may use this information when it calculates a bill segment for the contract if the rate has SQ (service quantity) rules or register rules.

The following fields display:

Effective Date The date this quantity becomes effective. The contract's start date defaults.

Contract Quantity Type The type of contract quantity.

Contract Quantity The contract quantity.

The **Recurring Charge** section contains effective-dated information that defines the recurring charge amount used to calculate the contract's bill segments. This information is only displayed if the contract's contract type allows a recurring charge amount.

A contract's recurring charge amount history is updated by the system behind-the-scenes as part of the Start Service process. The budget recalculation process can also update this information. You should only have to access this section if you need to change a contract's recurring charge amount on some future date.

Billing uses recurring charge information to select the appropriate recurring charge when it calculates a bill segment for the contract. To modify a recurring charge, simply move to a field and change its value. To change the recurring charge amount effective on a given date, click the + button to insert a row, then fill in the information for each field. The following fields display:

Effective Date The date the charge becomes effective. The contract's start date defaults.

Note: Recurring charge changes during a bill period. Only one recurring charge will be used on any bill segment. The system selects the recurring charge value effective on the end date of the bill segment.

Recurring Charge Amount The recurring charge amount.

Contract - Miscellaneous

The miscellaneous page contains information that describes miscellaneous information about the contract. Use **Customer Management, Contract, Misc** to open this page.

Description of Page

Start Requested By displays the information entered in the Requested By field on the Start Stop page when this contract was started.

Stop Requested By displays the information entered in the Requested By field on the Start Stop page when this contract was stopped.

Select the **SIC Code** associated with the customer. This is typically only used for commercial and industrial customers.

Enter a **Business Activity** associated with the customer. This can be used to further define what types of activities are served by this contract.

The bill message grid contains information about the message(s) to appear on a contract's bill segments. Both one-time and permanent messages may be defined. This information is used when the system assembles messages to appear on an account's bill (as part of bill completion). Refer to The Source Of Bill Messages for more information. The following fields are displayed:

- For **Bill Message Code**, select a message code to appear on the customer's bill.
- For **Bill Message Type**, use `Temporary` to indicate the message should only be linked to the next bill produced for the account. Use `Permanent` if the message should appear on every bill. Note, a value of `Temporary` defaults.

Copyright © 2024, Oracle and/or its affiliates.

The terms and conditions (T&C) grid allows you to define the contract's T&C's. Refer to *Legal Terms and Conditions May Be Specified on Contracts* for more information. The following fields display:

- Use **Start Date** to define the date on which the T&C becomes effective. The contract's start date defaults.
- Use **End Date** to define the date on which the T&C expires. This field need only be specified if the T&C expires on an explicit date.
- Use **Terms and Conditions** to define the type of T&C.

Automatic Contract Creation

You can now create contracts automatically for an account when the account is created. To enable this feature, you need to create an algorithm using the **C1-CRECONT** algorithm type and attach it to the **Post-Processing** system event of the respective account business object.

This algorithm is invoked when you create an account. A contract of each contract type listed in the **C1-CONTRACTTYPES** extendable lookup value is created automatically for the account. For example, an extendable lookup value has two contract types - CT1 and CT2 belonging to the D1 division. The system will then create two contracts - one using the CT1 contract type and another using the CT2 contract type. The status of the contract is set as mentioned in the extendable lookup value. In addition, the characteristics are defined for the contract as mentioned in the extendable lookup value. The contract start date and effective date of the contract characteristic (if any) is set to the account's set up date. If the closing date is specified for the account, the contract end date is set to the recalculated account closing date. However, the contract end date is not updated when the contract is created in the **Canceled** status.

It checks the value defined for the **Enforce division-specific validations** option type in the **C1_DIVFUNCT** feature configuration. If the **Enforce division-specific validations** option type is set to **Y**, it checks whether the account belongs to any division listed in the extendable lookup value. If the account belongs to a division listed in the extendable lookup value, then it automatically creates the contract of the respective contract type for the account. However, if the account does not belong to any division listed in the extendable lookup value, then it does not create the contracts for the account automatically.

This algorithm contains the following parameter:

- **Extendable Lookup Value** - Used to specify the value of the **C1-CONTRACTTYPES** extendable lookup.

> **Note:** You must specify a valid extendable lookup value which is in the **Active** status.

For more information about the **C1-CONTRACTTYPES** extendable lookup, see Automatic Contract Creation Extendable Lookup on page 73.

Automatic Contract Creation Extendable Lookup

The **Extendable Lookup Query** screen allows you to search for an extendable lookup using various search criteria. It contains the following zone:

- Extendable Lookup Search on page 73

Extendable Lookup Search

The **Extendable Lookup Search** zone allows you to search for an extendable lookup using various search criteria. It contains the following two sections:

- **Search Criteria** - The **Search Criteria** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Business Object	Used to search for a particular extendable lookup business object.	No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
Description	Used to search extendable lookup business objects with a particular description.	No

Note: You must specify at least one search criterion while searching for an extendable lookup.

- **Search Results** - On clicking the **Search** button, the search results appear based on the specified search criteria. The **Search Results** section contains the following columns:

Column Name	Column Description
Business Object	Displays the extendable lookup business object.
Description	Displays the description of the extendable lookup business object.
	Note: It has a link. On clicking the link, the **Extendable Lookup** screen appears where you can view the details of the respective extendable lookup.

Related Topics

For more information on...	See...
How to search for the **C1-CONTRACTTYPES** extendable lookup	Searching for the C1-CONTRACTTYPES Extendable Lookup on page 74

Searching for the C1-CONTRACTTYPES Extendable Lookup

Procedure

To search for the **C1-CONTRACTTYPES** extendable lookup:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **E** and then click **Extendable Lookup**.
   The **Extendable Lookup Query** screen appears.
3. Type `C1-CONTRACTTYPES` in the **Business Object** field.

   Note: ORMB search engine supports wildcard search, where you can substitute the percentage (%) symbol as a stand in for any word or letter in a search criteria. You can use the '%' wildcard character in all input fields except the date and ID fields. The '%' wildcard character is suffixed automatically at the end of the partial search criteria. Therefore, you may or may not specify the wildcard character at the end of the partial search criteria. However, you have to prefix the wildcard character manually wherever required.

4. Click **Search**.
   The **C1-CONTRACTTYPES** extendable lookup appears in the **Search Results** section.

Related Topics

For more information on...	See...
Extendable Lookup Query screen	Automatic Contract Creation Extendable Lookup on page 73
Extendable Lookup Search zone	Extendable Lookup Search on page 73

Copyright © 2024, Oracle and/or its affiliates.

Viewing the Values of the C1-CONTRACTTYPES Extendable Lookup

Procedure

To view the values of the **C1-CONTRACTTYPES** extendable lookup:

1. Search for the **C1-CONTRACTTYPES** extendable lookup in the **Extendable Lookup Query** screen.
2. In the **Search Results** section, click the link in the **Description** column corresponding to the **C1-CONTRACTTYPES** extendable lookup whose details you want to view.
   The **Extendable Lookup** screen appears.
3. View the values of the **C1-CONTRACTTYPES** extendable lookup in the **Extendable Lookup Value List** zone.

Related Topics

For more information on...	See...
How to search for the **C1-CONTRACTTYPES** extendable lookup	Searching for the C1-CONTRACTTYPES Extendable Lookup on page 74
Extendable Lookup screen	Extendable Lookup (Automatic Contract Creation) on page 77
Extendable Lookup Value List zone	Extendable Lookup Value List on page 78

Defining a Value for the C1-CONTRACTTYPES Extendable Lookup

Prerequisites

To define a value for the **C1-CONTRACTTYPES** extendable lookup, you should have:

- Divisions defined in the application
- Contract types defined for the required division in the application
- Characteristic types defined in the application (where the characteristic entity is set to **Contract**)

Procedure

To define a value for the **C1-CONTRACTTYPES** extendable lookup:

1. Search for the **C1-CONTRACTTYPES** extendable lookup in the **Extendable Lookup Query** screen.
2. In the **Search Results** section, click the link in the **Description** column corresponding to the **C1-CONTRACTTYPES** extendable lookup in which you want to add a value.
   The **Extendable Lookup** screen appears.
3. Click the **Add** button in the **Page Title** area of the **Extendable Lookup** screen.

   The **Extendable Lookup Value** screen appears. It contains the following sections:

   - **Main** - Used to specify basic details about the extendable lookup value. It contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Extendable Lookup Value	Used to specify the value for the extendable lookup.	Yes
Description	Used to specify the description for the extendable lookup value.	Yes
Override Description	Used when you want to override the description of the extendable lookup value.	No
Detailed Description	Used to specify addition information about the extendable lookup value.	No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
Status	Used to indicate the status of the extendable lookup value. The valid values are: • Active • Inactive **Note:** By default, the **Active** status is selected.	No

- **Contract Information** - Used to indicate the contract type using which the contract should be automatically created for accounts belonging to a particular division. It contains the following fields in a grid:

Field Name	Field Description	Mandatory (Yes or No)
Contract Characteristics	On clicking the **Add** icon, you can define characteristics for the contract which is automatically created on the account creation.	Not applicable
Division	Used to indicate the division whose accounts' contract must be automatically created.	Yes (Conditional) **Note:** This field is required when you are adding contract information in the extendable lookup value.
Contract Type	Used to indicate the contract type using which you want to create the contract. **Note:** The **Search** icon appears corresponding to this field. On clicking the **Search** icon, the **Contract Type Search** window appears.	Yes (Conditional) **Note:** This field is required when you are adding contract information in the extendable lookup value.
Contract Status	Used to indicate the status in which the contract must be created. The valid values are: • Active • Canceled • Incomplete • Pending Start • Pending Stop	Yes (Conditional) **Note:** This field is required when you are adding contract information in the extendable lookup value.

4. Enter the required details in the **Main** section.
5. Enter the details, such as the contract type, contract status and characteristics, using which the contract should be automatically created for accounts belonging to a particular division.
6. If you want to define a characteristic for contracts which are automatically created for accounts belonging a division, click the **Add** icon corresponding to the required division and contract type combination.

   A grid appears with the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Characteristic Type	Used to indicate the characteristic type.	Yes (Conditional)

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: The list includes only those characteristic types where the characteristic entity is set to **Contract**.	**Note:** This field is required when you are defining a characteristic for a contract which is automatically created.
Characteristic Value	Used to specify the value for the characteristic type.	Yes (Conditional)
	Note: On specifying the value for a predefined characteristic type, the description of the characteristic value appears corresponding to the **Characteristic Value** field.	**Note:** This field is required when you are defining a characteristic for a contract which is automatically created.

7. Enter the required details in the **Characteristics** grid.

> **Note:** If you select a predefined characteristic type, the **Search** (🔍) icon appears corresponding to the **Characteristic Value** field. On clicking the **Search** icon, you can search for a predefined characteristic value.

8. If you want to define more than one characteristic for the contract, click the **Add** (+) icon in the **Characteristics** grid and *t*hen repeat step 7.

> **Note:** However, if you want to remove a characteristic from the **Characteristics** grid, click the **Delete** (🗑) icon corresponding to the characteristic.

9. If you want to create more than one contract for accounts belonging to a particular division, click the **Add** (+) icon and then repeat steps from 5 to 8.

> **Note:** However, if you want to remove a division and contract type combination from the **Contract Information** section, click the **Delete** (🗑) icon corresponding to the division and contract type combination.

10. Click **Save**.
    The extendable lookup value is defined.

Related Topics

For more information on...	See...
How to search for the **C1-CONTRACTTYPES**extendable lookup	Searching for the C1-CONTRACTTYPES Extendable Lookup on page 74
Extendable Lookup screen	Extendable Lookup (Automatic Contract Creation) on page 77

Extendable Lookup (Automatic Contract Creation)

The **Extendable Lookup** screen allows you to view the values of an extendable lookup and the details of each extendable lookup value. It also allows you to define, edit, copy, and delete an extendable lookup value. It contains the following zones:

- Extendable Lookup Value List on page 78
- Extendable Lookup Value on page 78

Copyright © 2024, Oracle and/or its affiliates.

Extendable Lookup Value List

The **Extendable Lookup Value List** zone lists the values of an extendable lookup that are already defined in the system. It contains the following columns:

Column Name	Column Description
Value	Displays the value of the extendable lookup.
Description	Displays the description of the extendable lookup value.
Owner	Indicates whether the extendable lookup value is created by the framework, product, or customer. The valid values are: • Base • Customer Modification • Framework
Edit	On clicking the **Edit** (✎) icon, the **Extendable Lookup Value** screen appears where you can edit the details of the extendable lookup value.
Duplicate	On clicking the **Duplicate** (⧉) icon, the **Extendable Lookup Value** screen appears where you can define a new extendable lookup value using an existing extendable lookup value.
Delete	On clicking the **Delete** (🗑) icon, you can delete the extendable lookup value.
Detailed Description	Displays additional information about the extendable lookup value.

On clicking the **Broadcast** (📡) icon corresponding to an extendable lookup value, the **Extendable Lookup Value** zone appears with the details of the respective extendable lookup value.

Related Topics

For more information on...	See...
How to edit a value of the **C1-CONTRACTTYPES** extendable lookup	Editing a Value of the C1-CONTRACTTYPES Extendable Lookup on page 80
How to copy a value of the **C1-CONTRACTTYPES** extendable lookup	Copying a Value of the C1-CONTRACTTYPES Extendable Lookup on page 84
How to delete a value of the **C1-CONTRACTTYPES** extendable lookup	Deleting a Value of the C1-CONTRACTTYPES Extendable Lookup on page 83
How to view the details of the **C1-CONTRACTTYPES** extendable lookup value	Viewing the Details of the C1-CONTRACTTYPES Extendable Lookup Value on page 80

Extendable Lookup Value

The **Extendable Lookup Value** zone displays the details of the extendable lookup value. It contains the following sections:

- **Main** - Displays basic information about the extendable lookup value. It contains the following fields:

Field Name	Field Description
Extendable Lookup Value	Displays the value of the extendable lookup.
Description	Displays the description of the extendable lookup value.
Override Description	Displays the override description of the extendable lookup value.

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description
Detailed Description	Displays addition information about the extendable lookup value.
Status	Indicates the status of the extendable lookup value. The valid values are: • Active • Inactive

- **Contract Information** - Lists the contract types using which the contracts should be automatically created for accounts belonging to a particular division. It contains the following columns:

Column Name	Column Description
Division	Indicates the division whose accounts' contract must be automatically created.
Contract Type	Indicates the contract type using which the contract must be created.
Contract Status	Indicates the status in which the contract must be created. The valid values are: • Active • Canceled • Incomplete • Pending Start • Pending Stop
Contract Characteristics	On clicking the **Add** ( ) icon, you can view the characteristics which must be defined for the contract on creation.

In addition, a grid appears when you click the **Add** ( ) icon corresponding to a division and contract type combination. It contains the following columns:

Column Name	Column Description
Characteristic Type	Indicates the characteristic type.
Characteristic Value	Displays the value of the characteristic type.

- **Record Actions** - This section contains the following buttons:

Button Name	Button Description
Edit	Used to edit the details of the extendable lookup value.
Delete	Used to delete the extendable lookup value.
Duplicate	Used to create a new extendable lookup value using an existing extendable lookup value.

- **Record Information** - This section contains the following fields:

Field Name	Field Description
Business Object	Indicates the business object using which the extendable lookup value is created. In addition, a context menu appears corresponding to this field which helps in navigating to other screens in the application. **Note:** It has a link. On clicking the link, the **Business Object** screen appears where you can view the details of the respective business object.

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description
Owner	Indicates whether the extendable lookup value is created by the framework, product, or customer. The valid values are: • Base • Customer Modification • Framework

By default, the **Extendable Lookup Value** zone does not appear in the **Extendable Lookup** screen. It appears when you click the **Broadcast** (📶) icon corresponding to an extendable lookup value in the **Extendable Lookup Value List** zone.

Related Topics

For more information on...	See...
How to edit a value of the **C1-CONTRACTTYPES** extendable lookup	Editing a Value of the C1-CONTRACTTYPES Extendable Lookup on page 80
How to copy a value of the **C1-CONTRACTTYPES** extendable lookup	Copying a Value of the C1-CONTRACTTYPES Extendable Lookup on page 84
How to delete a value of the **C1-CONTRACTTYPES** extendable lookup	Deleting a Value of the C1-CONTRACTTYPES Extendable Lookup on page 83
How to view the details of the **C1-CONTRACTTYPES** extendable lookup value	Viewing the Details of the C1-CONTRACTTYPES Extendable Lookup Value on page 80

Viewing the Details of the C1-CONTRACTTYPES Extendable Lookup Value

Procedure

To view the details of the **C1-CONTRACTTYPES** extendable lookup value:

1. Search for the **C1-CONTRACTTYPES** extendable lookup in the **Extendable Lookup Query** screen.
2. In the **Search Results** section, click the link in the **Description** column corresponding to the **C1-CONTRACTTYPES** extendable lookup whose values you want to view.
   The **Extendable Lookup** screen appears.
3. In the **Extendable Lookup Value List** zone, click the **Broadcast** (📶) icon corresponding to an extendable lookup value whose details you want to view.
   The **Extendable Lookup Value** zone appears.
4. View the details of the extendable lookup value in the **Extendable Lookup Value** zone.

Related Topics

For more information on...	See...
How to search for the **C1-CONTRACTTYPES** extendable lookup	Searching for the C1-CONTRACTTYPES Extendable Lookup on page 74
Extendable Lookup screen	Extendable Lookup (Automatic Contract Creation) on page 77
Extendable Lookup Value List zone	Extendable Lookup Value List on page 78
Extendable Lookup Value zone	Extendable Lookup Value on page 78

Editing a Value of the C1-CONTRACTTYPES Extendable Lookup

Prerequisites

Copyright © 2024, Oracle and/or its affiliates.

To edit a value of the **C1-CONTRACTTYPES** extendable lookup, you should have:

- Divisions defined in the application
- Contract types defined for the required division in the application
- Characteristic types defined in the application (where the characteristic entity is set to **Contract**)

Procedure

To edit a value of the **C1-CONTRACTTYPES** extendable lookup:

1. Search for the **C1-CONTRACTTYPES** extendable lookup in the **Extendable Lookup Query** screen.
2. In the **Search Results** section, click the link in the **Description** column corresponding to the **C1-CONTRACTTYPES** extendable lookup whose value you want to edit.
   The **Extendable Lookup** screen appears.
3. In the **Extendable Lookup Value List** zone, click the **Edit** (✎) icon in the **Edit** column corresponding to the extendable lookup value whose details you want to edit.

   The **Extendable Lookup Value** screen appears. It contains the following sections:

   - **Main** - Used to specify basic details about the extendable lookup value. It contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Extendable Lookup Value	Displays the extendable lookup value.	Not applicable
Description	Used to specify the description for the extendable lookup value.	Yes
Override Description	Used when you want to override the description of the extendable lookup value.	No
Detailed Description	Used to specify addition information about the extendable lookup value.	No
Status	Used to indicate the status of the extendable lookup value. The valid values are: • Active • Inactive	No
Owner	Indicates whether the extendable lookup value is created by the framework, product, or customer. The valid values are: • Base • Customer Modification • Framework	Not applicable

   - **Contract Information** - Used to indicate the contract type using which the contract should be automatically created for accounts belonging to a particular division. It contains the following fields in a grid:

Field Name	Field Description	Mandatory (Yes or No)
Contract Characteristics	On clicking the **Add** (⊞) icon, you can define characteristics for the contract which is automatically created on the account creation.	Not applicable
Division	Used to indicate the division whose accounts' contract must be automatically created.	Yes (Conditional)

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
		Note: This field is required when you are adding contract information in the extendable lookup value.
Contract Type	Used to indicate the contract type using which you want to create the contract.	Yes (Conditional)
	Note: The **Search** (🔍) icon appears corresponding to this field. On clicking the **Search** icon, the **Contract Type Search** window appears.	**Note:** This field is required when you are adding contract information in the extendable lookup value.
Contract Status	Used to indicate the status in which the contract must be created. The valid values are: • Active • Canceled • Incomplete • Pending Start • Pending Stop	Yes (Conditional)
		Note: This field is required when you are adding contract information in the extendable lookup value.

Tip: Alternatively, you can edit the details of an extendable lookup value by clicking the **Edit** button in the **Extendable Lookup Value** zone.

4. Modify the required details in the **Main** section.
5. If required, modify the details, such as the contract type, contract status and characteristics, using which the contract should be automatically created for accounts belonging to a particular division.
6. If you want to define a characteristic for contracts which are automatically created for accounts belonging a division, click the **Add** (⊞) icon corresponding to the required division and contract type combination.

   A grid appears with the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Characteristic Type	Used to indicate the characteristic type.	Yes (Conditional)
	Note: The list includes only those characteristic types where the characteristic entity is set to **Contract**.	**Note:** This field is required when you are defining a characteristic for a contract which is automatically created.
Characteristic Value	Used to specify the value for the characteristic type.	Yes (Conditional)
	Note: On specifying the value for a predefined characteristic type, the description of the characteristic value appears corresponding to the **Characteristic Value** field.	**Note:** This field is required when you are defining a characteristic for a contract which is automatically created.

7. Enter the required details in the **Characteristics** grid.

Note: If you select a predefined characteristic type, the **Search** (🔍) icon appears corresponding to the **Characteristic Value** field. On clicking the **Search** icon, you can search for a predefined characteristic value.

8. If you want to define more than one characteristic for the contract, click the **Add** (+) icon in the **Characteristics** grid and then repeat step 7.

Copyright © 2024, Oracle and/or its affiliates.

> **Note:** However, if you want to remove a characteristic from the **Characteristics** grid, click the **Delete** (🗑) icon corresponding to the characteristic.

9. If you want to create more than one contract for accounts belonging to a particular division, click the **Add** (+) icon and then repeat steps from 5 to 8.

> **Note:** However, if you want to remove a division and contract type combination from the **Contract Information** section, click the **Delete** (🗑) icon corresponding to the division and contract type combination.

10. Click **Save**.
    The changes made to the extendable lookup value are saved.

Related Topics

For more information on...	See...
How to search for the **C1-CONTRACTTYPES**extendable lookup	Searching for the C1-CONTRACTTYPES Extendable Lookup on page 74
Extendable Lookup screen	Extendable Lookup (Automatic Contract Creation) on page 77
Extendable Lookup Value List zone	Extendable Lookup Value List on page 78
Extendable Lookup Value zone	Extendable Lookup Value on page 78

Deleting a Value of the C1-CONTRACTTYPES Extendable Lookup

Procedure

To delete a value of the **C1-CONTRACTTYPES** extendable lookup:

1. Search for the **C1-CONTRACTTYPES** extendable lookup in the **Extendable Lookup Query** screen.
2. In the **Search Results** section, click the link in the **Description** column corresponding to the **C1-CONTRACTTYPES** extendable lookup whose value you want to delete.
   The **Extendable Lookup** screen appears.
3. In the **Extendable Lookup Value List** zone, click the **Delete** (🗑) icon in the **Delete** column corresponding to the extendable lookup value that you want to delete.
   A message appears confirming whether you want to delete the extendable lookup value.

> **Tip:** Alternatively, you can delete an extendable lookup value by clicking the **Delete** button in the **Extendable Lookup Value** zone.

4. Click **OK**.
   The extendable lookup value is deleted.

Related Topics

For more information on...	See...
How to search for the **C1-CONTRACTTYPES**extendable lookup	Searching for the C1-CONTRACTTYPES Extendable Lookup on page 74
Extendable Lookup screen	Extendable Lookup (Automatic Contract Creation) on page 77
Extendable Lookup Value List zone	Extendable Lookup Value List on page 78
Extendable Lookup Value zone	Extendable Lookup Value on page 78

Copyright © 2024, Oracle and/or its affiliates.

Copying a Value of the C1-CONTRACTTYPES Extendable Lookup

Instead of creating an extendable lookup value from scratch, you can create a new extendable lookup value using an existing extendable lookup value. This is possible through copying an extendable lookup value. On copying an extendable lookup value, the details including the contract information are copied to the new extendable lookup value. You can then edit the details, if required.

Prerequisites

To copy a value of the **C1-CONTRACTTYPES** extendable lookup, you should have:

- Extendable lookup value (whose copy you want to create) defined in the application
- Divisions defined in the application
- Contract types defined for the required division in the application
- Characteristic types defined in the application (where the characteristic entity is set to **Contract**)

Procedure

To copy a value of the **C1-CONTRACTTYPES** extendable lookup:

1. Search for the **C1-CONTRACTTYPES** extendable lookup in the **Extendable Lookup Query** screen.
2. In the **Search Results** section, click the link in the **Description** column corresponding to the **C1-CONTRACTTYPES** extendable lookup whose value you want to copy.
   The **Extendable Lookup** screen appears.
3. In the **Extendable Lookup Value List** zone, click the **Duplicate** (⧉) icon in the **Duplicate** column *c*orresponding to the extendable lookup value whose copy you want to create.

   The **Extendable Lookup Value** screen appears. It contains the following sections:

   - **Main** - Used to specify basic details about the extendable lookup value. It contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Extendable Lookup Value	Used to specify the value for the extendable lookup.	Yes
Description	Used to specify the description for the extendable lookup value.	Yes
Override Description	Used when you want to override the description of the extendable lookup value.	No
Detailed Description	Used to specify addition information about the extendable lookup value.	No
Status	Used to indicate the status of the extendable lookup value. The valid values are: • Active • Inactive	No

   - **Contract Information** - Used to indicate the contract type using which the contract should be automatically created for accounts belonging to a particular division. It contains the following fields in a grid:

Field Name	Field Description	Mandatory (Yes or No)
Contract Characteristics	On clicking the **Add** (⊞) icon, you can define characteristics for the contract which is automatically created on the account creation.	Not applicable
Division	Used to indicate the division whose accounts' contract must be automatically created.	Yes (Conditional)

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
		Note: This field is required when you are adding contract information in the extendable lookup value.
Contract Type	Used to indicate the contract type using which you want to create the contract.	Yes (Conditional)
	Note: The **Search** (🔍) icon appears corresponding to this field. On clicking the **Search** icon, the **Contract Type Search** window appears.	**Note:** This field is required when you are adding contract information in the extendable lookup value.
Contract Status	Used to indicate the status in which the contract must be created. The valid values are: • Active • Canceled • Incomplete • Pending Start • Pending Stop	Yes (Conditional)
		Note: This field is required when you are adding contract information in the extendable lookup value.

Tip: Alternatively, you can copy an extendable lookup value by clicking the **Duplicate** button in the **Extendable Lookup Value** zone.

4. Enter the required details in the **Main** section.
5. Enter the details, such as the contract type, contract status and characteristics, using which the contract should be automatically created for accounts belonging to a particular division.
6. If you want to define a characteristic for contracts which are automatically created for accounts belonging a division, click the **Add** (⊞) icon corresponding to the required division and contract type combination.

   A grid appears with the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Characteristic Type	Used to indicate the characteristic type.	Yes (Conditional)
	Note: The list includes only those characteristic types where the characteristic entity is set to **Contract**.	**Note:** This field is required when you are defining a characteristic for a contract which is automatically created.
Characteristic Value	Used to specify the value for the characteristic type.	Yes (Conditional)
	Note: On specifying the value for a predefined characteristic type, the description of the characteristic value appears corresponding to the **Characteristic Value** field.	**Note:** This field is required when you are defining a characteristic for a contract which is automatically created.

7. Enter the required details in the **Characteristics** grid.

Note: If you select a predefined characteristic type, the **Search** (🔍) icon appears corresponding to the **Characteristic Value** field. On clicking the **Search** icon, you can search for a predefined characteristic value.

8. If you want to define more than one characteristic for the contract, click the **Add** (+) icon in the **Characteristics** grid and then repeat step 7.

Copyright © 2024, Oracle and/or its affiliates.

> **Note:** However, if you want to remove a characteristic from the **Characteristics** grid, click the **Delete** (🗑) icon corresponding to the characteristic.

9. If you want to create more than one contract for accounts belonging to a particular division, click the **Add** (+) icon and then repeat steps from 5 to 8.

> **Note:** However, if you want to remove a division and contract type combination from the **Contract Information** section, click the **Delete** (🗑) icon corresponding to the division and contract type combination.

10. Click **Save**.
    The new extendable lookup value is defined.

Related Topics

For more information on...	See...
How to search for the **C1-CONTRACTTYPES**extendable lookup	Searching for the C1-CONTRACTTYPES Extendable Lookup on page 74
Extendable Lookup screen	Extendable Lookup (Automatic Contract Creation) on page 77
Extendable Lookup Value List zone	Extendable Lookup Value List on page 78
Extendable Lookup Value zone	Extendable Lookup Value on page 78

Person View

The **Person View** screen allows you to view the person details. It contains the following zones:

- Person Information on page 86
- Person Tree on page 88
- Person Characteristics on page 88
- Person Relationship on page 88
- Person Relationship Characteristics on page 89

Person Information

The **Person Information** zone displays information about the person. It contains the following sections:

- **Main** - This section displays the basic details of the person. It contains the following fields:

Field Name	Field Description
Person Information	Displays information about the person. In addition, a context menu appears corresponding to this field which helps in navigating to other screens in the application.
	Note: It has a link. On clicking the link, the **Person** screen appears with the details of the respective person. The person information string appears when an algorithm defined using the **PERS-INFO** algorithm type is attached to the **Person Information** system event in the **Algorithms** tab of the **Installation Options - Framework** screen.
Person Type	Indicates the type of person.

Copyright © 2024, Oracle and/or its affiliates.

- **Person IDs** - This section lists various types of identification associated with the person. It contains the following columns:

Column Name	Column Description
Person Identifier Type	Indicates the person identifier type.
Person Identifier	Displays the value of the person identifier type.

- **Contact Information** - This section displays the contact details of the person. It contains the following fields:

Field Name	Field Description
Address Line 1	Displays the first line of person's address. It may contain details, such as house number and apartment name.
Address Line 2	Displays the second line of person's address. It may contain details, such as street name.
Address Line 3	Displays the third line of person's address. It may contain the landmark details.
City	Displays the city where the person is located.
State	Indicates the state where the city is located.
Country	Indicates the country where the state is located.
County	Displays the county where the person is located.

Note: The above mentioned fields appear in the **Contact Information** section when the respective data is available for the person.

In addition, this section contains the following columns:

Column Name	Column Description
Phone Type	Indicates the type of phone number.
Phone Number	Displays the telephone number of the person.

- **Seasonal Address Information** - This section lists seasonal address of the person which is maintained for the predefined period. It contains the following columns:

Column Name	Column Description
Address Line 1	Displays the first line of person's seasonal address. It may contain details, such as house number and apartment name.
Address Line 2	Displays the second line of person's seasonal address. It may contain details, such as street name.
Address Line 3	Displays the third line of person's seasonal address. It may contain the landmark details.
City	Displays the city where the seasonal address is located.

Related Topics

For more information on...	See...
Person View screen	Person View on page 86
How to view the details of a person	Viewing the Person Details on page 90

Copyright © 2024, Oracle and/or its affiliates.

Person Tree

The **Person Tree** zone lists the following in a tree structure:

- Information string of accounts to which the person is linked.

  Note: The account information string appears when an algorithm defined using the **C1-AC-INFO** algorithm type is attached to the **Account Information** system event in the **Algorithms** tab of the **Installation Options - Framework** screen.

- Information string of any other persons linked to the account

  Note: The person information string appears when an algorithm defined using the **PERS-INFO** algorithm type is attached to the **Person Information** system event in the **Algorithms** tab of the **Installation Options - Framework** screen.

Each node in the tree has a link. On clicking the link, you can navigate to the respective screen.

Related Topics

For more information on...	See...
Person View screen	Person View on page 86
How to view the details of a person	Viewing the Person Details on page 90

Person Characteristics

The **Person Characteristics** zone lists the characteristics of the person. It contains the following columns:

Column Name	Column Description
Effective Date	Displays the date from when the characteristic is effective for the person.
Characteristic Type	Indicates the characteristic type.
Characteristic Value	Displays the value of the characteristic type.
Description	Displays the description of the characteristic value.
	Note: The data appears in this column when the type of characteristic value is set to **Predefined Value** or **Foreign Key Value**.

You can filter the list using various search criteria (such as, **Characteristic Type** and **Effective Date**) available in the **Filter** area. By default, the **Filter** area is hidden. You can view the **Filter** area by clicking the **Filters** (▽) icon in the upper right corner of this zone.

You can change the layout of this zone. For more information on how to change the layout, see Changing the Layout.

Related Topics

For more information on...	See...
Person View screen	Person View on page 86
How to view the details of a person	Viewing the Person Details on page 90

Person Relationship

The **Person Relationship** zone lists child persons included in the person's hierarchy. It contains the following columns:

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
Child Person	Indicates the child person.
Start Date	Displays the date from when the child person is related to the parent person.
End Date	Displays the date till when the child person is related to the parent person.
Description (Person1=> Person2)	Indicates how the parent person is related to the child person.
Description (Person2=> Person1)	Indicates how the child person is related to the parent person.
Financial Relationship	Indicates whether the child person is financially responsible for the parent person. The valid values are: • Yes • No

You can filter the list using various search criteria (such as, **Start Date**, **End Date**, and **Relationship Type**) available in the **Filter** area. By default, the **Filter** area is hidden. You can view the **Filter** area by clicking the **Filters** (▽) icon in the upper right corner of this zone.

You can change the layout of this zone. For more information on how to change the layout, see Changing the Layout.

On clicking the **Broadcast** (ᯤ) icon corresponding to the child person, the **Person Relationship Characteristics** zone appears.

Related Topics

For more information on...	See...
Person View screen	Person View on page 86
How to view the details of a person	Viewing the Person Details on page 90

Person Relationship Characteristics

The **Person Relationship Characteristics** zone lists the characteristics of the person to person relationship. It contains the following columns:

Column Name	Column Description
Effective Date	Displays the date from when the characteristic is effective for the person to person relationship.
Characteristic Type	Indicates the characteristic type.
Characteristic Value	Displays the value of the characteristic type.
Description	Displays the description of the characteristic value.
	Note: The data appears in this column when the type of characteristic value is set to **Predefined Value** or **Foreign Key Value**.

By default, the **Person Relationship Characteristics** zone does not appear in the **Person View** screen. It appears when you click the **Broadcast** (ᯤ) icon corresponding to the child person in the **Person Relationship** zone.

Related Topics

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
Person View screen	Person View on page 86
How to view the details of a person	Viewing the Person Details on page 90

Viewing the Person Details

Prerequisites

To view the details of a person, you should have:

- An algorithm defined using the **C1-AC-INFO** algorithm type and attached to the **Account Information** system event in the **Algorithms** tab of the **Installation Options - Framework** screen
- An algorithm defined using the **PERS-INFO** algorithm type and attached to the **Person Information** system event in the **Algorithms** tab of the **Installation Options - Framework** screen

Procedure

To view the details of a person:

1. Do either of the following:

If you want to...	Then
Access the Person View screen through the Customer 360° View screen	**a.** Search for the person in the **Customer 360° View** screen. **b.** In the **Search Results** section, click the **Show Context Menu** (▾) icon corresponding to the person whose *d*etails you want to view. A menu appears. **c.** Click the **Go To Person View** menu option.
Access the Person View screen through the Person screen	**a.** Click the **Menu** link in the **Application** toolbar. A list appears. **b.** From the **Main** menu, select **Person Information** and then click **Person**. A sub-menu appears. **c.** Click the **Search** option from the **Person** sub-menu. The **Person Search** window appears. **d.** Search for the person in the **Person Search** window. **e.** In the **Search Results** section, click the person whose details you want to view. The **Person** screen appears. **f.** Click the **Show Context Menu** (▾) icon corresponding to the **Person ID** field. A menu appears. **g.** Click the **Go To Person View** menu option.
Access the Person View screen through the Current Context zone on the dashboard	**a.** In the **Current Context** zone, click the **Show Context Menu** (▾) icon corresponding to the person information string. A menu appears. **b.** Click the **Go To Person View** menu option.

The **Person View** screen appears. It contains the following zones:

- Person Information

Copyright © 2024, Oracle and/or its affiliates.

- Person Tree
- Person Characteristics
- Person Relationship
- Person Relationship Characteristics

2. View the details of the person in the **Person Information** zone.
3. View the accounts linked to the person in the **Person Tree** zone.
4. View the characteristics of the person in the **Person Characteristics** zone.
5. View the child persons included in the person's hierarchy in the **Person Relationship** zone.
6. View the characteristics of the person to person relationship in the **Person Relationship Characteristics** zone.

Related Topics

For more information on...	**See...**
Person View screen	Person View on page 86
Person Information zone	Person Information on page 86
Person Tree zone	Person Tree on page 88
Person Characteristics zone	Person Characteristics on page 88
Person Relationship zone	Person Relationship on page 88
Person Relationship Characteristics zone	Person Relationship Characteristics on page 89

Account View

The **Account View** screen allows you to view the account details. It contains the following zones:

- Account Information on page 91
- Account Tree on page 93
- Account Characteristics on page 93

Account Information

The **Account Information** zone displays basic details of the account including persons linked to the account and various types of identification associated with the account. It contains the following sections:

- **Main** - This section provides basic information about the account. It contains the following fields:

Field Name	**Field Description**
Account Information	Displays information about the account. In addition, a context menu appears corresponding to this field which helps in navigating to other screens in the application.

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description
	Note: It has a link. On clicking the link, the **Account** screen appears with the details of the respective account. The account information string appears when an algorithm defined using the **C1-AC-INFO** algorithm type is attached to the **Account Information** system event in the **Algorithms** tab of the **Installation Options - Framework** screen.
Customer Class	Indicates the customer class to which the account belongs.
Division	Indicates the division to which the account belongs.
Eligible for Repricing	Indicates whether the account is eligible for repricing.
Bill Cycle	Indicates the bill cycle defined for the account. **Note:** This field appears when the bill cycle is defined for the account.
Set Up Date	Displays the date when the account is created in the system.
Invoice Currency	Indicates the invoice currency of the account.

- **Account Identifiers** - This section lists various types of identification associated with the account. It contains the following columns:

Column Name	Column Description
Account Identifier Type	Indicates the account identifier type.
Account Identifier	Displays the value of the account identifier type.

- **Account Persons** - This section provides details of the persons linked to the account. It contains the following columns:

Column Name	Column Description
Person Information	Displays information about the person. In addition, this column has a context menu which helps in navigating to other screens in the application. **Note:** It has a link. On clicking the link, the **Person** screen appears where you can view the details of the respective person. The person information string appears when an algorithm defined using the **PERS-INFO** algorithm type is attached to the **Person Information** system event in the **Algorithms** tab of the **Installation Options - Framework** screen.
Relationship Type	Indicates how the person is related to the account.
Main Customer	Indicates whether the person is a main customer of the account. The valid values are: • True • False
Financially Responsible	Indicates whether the person is financially responsible for the account. The valid values are:

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
	• True • False

Related Topics

For more information on...	See...
Account View screen	Account View on page 91
How to view the details of an account	Viewing the Account Details on page 94

Account Tree

The **Account Tree** zone lists the following in a tree structure:

- Information string of persons linked to the account

  Note: The person information string appears when an algorithm defined using the **PERS-INFO** algorithm type is attached to the **Person Information** system event in the **Algorithms** tab of the **Installation Options - Framework** screen.

- Information string of any other accounts to which the person is linked.

  Note: The account information string appears when an algorithm defined using the **C1-AC-INFO** algorithm type is attached to the **Account Information** system event in the **Algorithms** tab of the **Installation Options - Framework** screen.

Each node in the tree has a link. On clicking the link, you can navigate to the respective screen.

Related Topics

For more information on...	See...
Account View screen	Account View on page 91
How to view the details of an account	Viewing the Account Details on page 94

Account Characteristics

The **Account Characteristics** zone lists the characteristics of the account. It contains the following columns:

Column Name	Column Description
Effective Date	Displays the date from when the characteristic is effective for the account.
Characteristic Type	Indicates the characteristic type.
Characteristic Value	Displays the value of the characteristic type.
Description	Displays the description of the characteristic value.
	Note: The data appears in this column when the type of characteristic value is set to **Predefined Value** or **Foreign Key Value**.

You can filter the list using various search criteria (such as, **Characteristic Type** and **Effective Date**) available in the **Filter** area. By default, the **Filter** area is hidden. You can view the **Filter** area by clicking the **Filters** (▽) icon in the upper right corner of this zone.

You can change the layout of this zone. For more information on how to change the layout, see Changing the Layout.

Copyright © 2024, Oracle and/or its affiliates.

Related Topics

For more information on...	See...
Account View screen	Account View on page 91
How to view the details of an account	Viewing the Account Details on page 94

Viewing the Account Details

Prerequisites

To view the details of an account, you should have:

- An algorithm defined using the **C1-AC-INFO** algorithm type and attached to the **Account Information** system event in the **Algorithms** tab of the **Installation Options - Framework** screen
- An algorithm defined using the **PERS-INFO** algorithm type and attached to the **Person Information** system event in the **Algorithms** tab of the **Installation Options - Framework** screen

Procedure

To view the details of an account:

1. Do either of the following:

If you want to...	Then
Access the Account View screen through the Customer 360° View screen	**a.** Search for the account in the **Customer 360° View** screen. **b.** In the **Search Results** section, click the **Show Context Menu** (▾) icon corresponding to the account whose *d*etails you want to view. A menu appears. **c.** Click the **Go To Account View** menu option.
Access the Account View screen through the Account screen	**a.** Click the **Menu** link in the **Application** toolbar. A list appears. **b.** From the **Main** menu, select **Person Information** and then click **Account**. A sub-menu appears. **c.** Click the **Search** option from the **Account** sub-menu. The **Account Search** window appears. **d.** Search for the account in the **Account Search** window. **e.** In the **Search Results** section, click the account whose details you want to view. The **Account** screen appears. **f.** Click the **Show Context Menu** (▾) icon corresponding to the **Account ID** field. A menu appears. **g.** Click the **Go To Account View** menu option.
Access the Account View screen through the Current Context zone on the dashboard	**a.** In the **Current Context** zone, click the **Show Context Menu** (▾) icon corresponding to the account information string. A menu appears. **b.** Click the **Go To Account View** menu option.

The **Account View** screen appears. It contains the following zones:

Copyright © 2024, Oracle and/or its affiliates.

- Account Information
- Account Tree
- Account Characteristics

2. View the details of the account in the **Account Information** zone.
3. View the persons linked to the account in the **Account Tree** zone.
4. View the characteristics of the account in the **Account Characteristics** zone.

Related Topics

For more information on...	See...
Account View screen	Account View on page 91
Account Information zone	Account Information on page 91
Account Tree zone	Account Tree on page 93
Account Characteristics zone	Account Characteristics on page 93

Account Collection Summary

Oracle Revenue Management and Billing allows you to view an account's current balance, unpaid invoices, on account payments, bill graph and timelines for customer contact, bills and payments for an account on the **Account Collection Summary** screen.

Account Collection Query section enables user to search for the account by Account.

The **Account Collection Summary** screen consists of two tabs.

- **Main** - Used to search for accounts. It consists of following zones:
  - Search on page 95
- The **Collections** tab contains the following zones:
  - Account Information on page 97
  - Bill Graph on page 98
  - Timeline on page 98
  - View Statement Details on page 100
  - Administrative Contracts on page 98

Account Collection Summary - Main

The **Main** tab displays the basic details of the **Account Collection Summary**. It consists of the following zone:

- Search on page 95

Search

The **Search** zone allows you to search for accounts and collections in the system. It contains the following sections:

- **Search Criteria**

  - This section contains the following fields:

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
Division	Used to search for accounts which belong to a particular division.	No
Person Identifier Type	Used to search for accounts for which a particular person identifier type is defined.	Yes (Conditional) **Note:** If you specify the person identifier as a search criteria, you have to select the person identifier type.
Person Name	Used to specify the name of the person whose accounts you want to search.	Yes (Conditional) **Note:** If you specify the person class as a search criteria, you have to specify the person name.
Customer Class	Used to specify the customer class of the account.	No
Collection Class	Used to indicate the collection class to which the account belongs.	Yes
Person Identifier	Used to search for an account with a particular person identifier.	Yes (Conditional) **Note:** If you specify the person identifier type as a search criteria, you have to select the person identifier .
Account ID	Used to specify the account ID.	Yes

- **Search Results** - This section contains the following columns:

Column Name	Column Description
Division	Used to search for accounts which belong to a particular division.
Account ID	Used to specify the account ID.
Invoice Currency	Used when you want to search accounts having a particular invoice currency.
Current Balance	Displays the amount of money in the account.
Person Identifier	Used to search for an account with a particular person identifier.
Person Name	Used to specify the name of the person whose accounts you want to search.
Person Identifier Type	Used to search for accounts for which a particular person identifier type is defined.
Person Name	Used to specify the name of the person whose accounts you want to search.
Person ID	Displays the person ID.
Customer Class	Used to specify the customer class of the account.

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
Collection Class	Used to indicate the collection class to which the account belongs.

Note: By default, the **Search Results** section does not appear. You have to click on**Search button**, after entering the required criteria.

Note: ORMB search engine supports wildcard search, where you can substitute the percentage (%) symbol as a stand in for any word or letter in a search criteria. You can use the '%' wildcard character in all input fields except the date and ID fields. The '%' wildcard character is suffixed automatically at the end of the partial search criteria. Therefore, you may or may not specify the wildcard character at the end of the partial search criteria. However, you have to prefix the wildcard character manually wherever required.

Related Topics

For more information on...	See...
Search zone	Search on page 95
How to search for accounts in accounts collection	Searching for Accounts in Accounts Collection\

Account Collection Summary - Collection

The **Collection** tab displays the collection details of an account . It consists of the following zones:

- Account Information on page 97
- Administrative Contracts on page 98
- Bill Graph on page 98
- Timeline on page 98
- Unpaid Bills on page 99
- View Statement Details on page 100

Account Information

The **Account Information** zone displays the details of an account or a person. This zone contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Account	Displays the additional account information. In addition, a context menu appears corresponding to this field which helps in navigating to other screens in the application. **Note:** It has a link. On clicking the link, the Accounts screen appears where you can view the details of the account.	Not applicable
Person Name	Displays the name of the person to whom the account belongs. In addition, a context menu appears corresponding to this field which helps in navigating to other screens in the application. **Note:** It has a link. On clicking the link, the Person screen appears where you can view the details of a person. .	Not applicable

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
Collection Class	Displays the collection class to which the account belongs.	Not applicable

Related Topics

For more information on...	See...
How to search for accounts and collections	Searching for Accounts in Accounts Collection\

Administrative Contracts

The **Administrative Contracts** zone displays the details of the contracts. This zone contains the following fields:

Field Name	Field Description
Contract ID	Used to indicate the contract whose contract information you want to update. **Note:** It has a link. On clicking the link, the **Contract** screen appears where you can view the details of the respective contract.
Contract Type	Used to indicate the type of the contract whose details you want to update.
Currency	Used to specify the currency in which the amount is charged.
Current Balance	Displays the amount of money in the account.

Related Topics

For more information on...	See...
How to search for accounts and collections	Searching for Accounts in Accounts Collection\

Bill Graph

The **Bill Graph** zone displays the graphical representation of the account details with the bill amount on y-axis and date on the x-axis.

Clicking on a particular bar will direct you to the **Bill** screen which displays all the details of the bill for the corresponding date.

The vertical bars are color coded as per below legend:

- Orange: Current Charge.
- Blue: Ending Balance.
- Green: Displays total freezable amount for the account.

Related Topics

For more information on...	See...
How to search for accounts and collections	Searching for Accounts in Accounts Collection\

Timeline

The **Timeline** zone displays the account information as per selected month and year.

Copyright © 2024, Oracle and/or its affiliates.

Displays the count of the customer contacts, bills and payments. Each point corresponding to a particular month-year and entity (customer contacts/bills/payments) are color coded as per below legend:

- Red: Displays total unpaid amount for the account.
- Blue: Displays total frozen amount for the account.
- Green: Displays total freezable amount for the account.
- Yellow: Displays total .....for the account.

Hovering the mouse over the color coded points displays additional information of the corresponding entity (customer contacts, bills and payments).

You can select the **Months** and **Years** fields and click**Search by Date** to view the entity (customer contacts, bills and payments) details for last 12 months from the selected month and year.

Note:

You can use the navigation buttons, such as **Previous 12 Months** ( ), **Previous Month** ( ), **Go To Today** ( ), **Next Month** ( ) and **Next 12 Months** ( ) to navigate between pages. You can also directly select the month and year using the Months, Years and Search by Date options.

Related Topics

For more information on...	See...
How to search for accounts and collections	Searching for Accounts in Accounts Collection\

Unpaid Bills

The **Unpaid Bills** zone lists and displays the details of the unpaid invoices. This zone contains the following fields:

Column Name	Column Description
Bill Date	Displays the date of invoice creation. The bill dates based on age are color coded as given below: • Red: Age is greater than 90 days. • Orange: Age is greater than 60 and less than or equal to 90. • Green: Age is greater than 30 and less than or equal to 60. • No highlight: Age is less than or equal to 30 or Unpaid Amount less than or equal to 0.
Bill ID	Displays the bill ID. **Note:** It has a link. On clicking the link, the **Bill** screen appears with the details of the respective bill.
Due Date	Displays the due date for the bill ID.
Days Past Due	Displays the no. of days past the due date.
Bill Status	Displays the status of the bill.
Invoice Currency	Displays the currency in which the amount is billed to the person.
Bill Amount	Displays the bill amount.
Unpaid Amount	Displays the unpaid amount of the bill.
Overdue Process ID	Displays the ID assigned for overdue processing. Overdue processing is a series of events (e.g., letters, To Do entries) meant to encourage an account holder to pay the debt.

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
Age less than 30	Displays the invoice amount for the invoice which is unpaid for less than 30 days from the date of creation.
Age 31 to 60	Displays the invoice amount for the invoice which is unpaid for 31-60 days from the date of creation.
Age 61 to 90	Displays the invoice amount for the invoice which is unpaid for 61-90 days from the date of creation.
Age greater than 90	Displays the invoice amount for the invoice which is unpaid for more than 90 days from the date of creation.

Note: You can search for the unpaid bills, by selecting a date in the **On or Before Bill Date** field. You can select the check-box corresponding to the **Show All Bills** field.

Related Topics

For more information on...	See...
How to search for accounts and collections	Searching for Accounts in Accounts Collection\

View Statement Details

The **View Statement Details** zone displays the details of the statement. It contains the following columns:

Column Name	Column Description
Coverage Period Start Date	Displays the start date of the coverage cycle.
Coverage Period End Date	Displays the end date of the coverage cycle.
Policy Number	Indicates the policy for which the bill segment is generated.
Plan/Fee Type	Indicates the plan code or fee type for which the bill segment is created.
External Membership ID	Indicates the membership to which the member belongs for whom the bill segment is generated.
Member's Primary ID	Displays the concatenated string consisting of primary identifier type and primary identifier of the member (for whom the bill segment is generated) which are separated by a colon (:).
Member Name	Displays the name of the member.
Coverage Tier	Indicates the coverage tier to which the membership belongs. The valid values are: • CHILDREN ONLY • EMPLOYEE & CHILD(REN) **Note:** The data appears in this column when the **Coverage Tier (C1CVTIER)** characteristic type is defined for the membership.
Status	Indicates the eligibility status of the member. The valid values are: • Active • Cobra • Retiree • Terminated

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
	Note: The data appears in this column when the **Eligibility Status Type (C1ELGSTA)** characteristic type is defined for the member.
Amount	Displays the bill segment amount.
Unpaid Amount	Displays the unpaid amount of the bill segment.

Note:

You can filter the list using various search criteria (such as, **Start Date** and **End Date**) available in the **Filter** area. By default, the **Filter** area is hidden. You can view the **Filter** area by clicking the **Filters** (▽) icon in the upper right corner of this zone.

On clicking the **View** (🔗 icon, the **Bill Segment** screen appears where you can view the details of the bill segment.

Related Topics

For more information on...	See...
Search zone	Search on page 95
How to search for accounts in accounts collection	Searching for Accounts in Accounts Collection\

Dashboard Portal

The Dashboard is a portal that always appears on the Oracle Revenue Management and Billing desktop. Its zones contain tools and data that are useful regardless of the object being displayed.

Note: User configurable. Refer to Each User Can Customize Which Zones Appear for information about a user can configure which zones appear.

Note: Global context. The system automatically refreshes the values saved in global context with information about the object that appears in Object Display Area. Various zones available in the dashboard portal have been designed to display data related to a person id or account id , which are populated by the base global context algorithm C1-GLBL-CTXT.

The contents of this section describe the zones that are available on this portal.

Note: Other zones exist. The zones described below are unique to Oracle Revenue Management and Billing. Please see Core Dashboard Zones for a description of other zones that can appear on the dashboard.

Current Context Zone

The **Current Context Zone** contains basic information about the customer on which you are working.

A maximum of three rows may appear:

- The first row contains a person context menu and the person's name. You can click on the person's name to transfer to the Person - Main page. Note, the person's name is formatted by a plug-in algorithm on the installation record. Refer to the base package's name format algorithm for an example. If you prefer different formatting logic, your system administrator should configure the system appropriately.

Copyright © 2024, Oracle and/or its affiliates.

- The second row contains an account context menu and the account's:
  - ID and check digit
  - The primary name of the main customer
  - Customer class

Customer Contact Zone

The **Customer Contact Zone** has three purposes:

- It shows the age of the **Last** customer contact associated with the person displayed in the Current Context zone. The word `Today` is shown if the last contact was on the current date. The word `Yesterday` is shown if the last contact was on the current date - 1 day. If neither is applicable, the number of days old is shown.
- It shows the name of the person who added this contact.
- It greatly simplifies the addition of a new customer contact. To add a new contact, simply select the **Type** of customer contact, enter a brief **Comment**, and press the **Add Contact** button.
- After the contact is added, the date / time of the new contact will be displayed in the **Last** area. You can use the Go To button to drill into the newly added contact where you can setup reminders or indicate the contact should remain open until future resolution.

Financial Information Zone

The **Financial Information Zone** is a grid that contains financial information related to the account. Clicking the hyperlink transfers you to the appropriate page.

Note: Rows are suppressed if the related data is blank for the person / account.

The following table lists the information that appears in the **Financial Information** zone:

Label	What Is Displayed	Drill Down Transfers You To
Current Balance	The account's current balance. If the account's current balance is greater than zero, a colored bar is displayed adjacent. This bar provides information about the age of the account's debt.	Clicking on the Arrears Bar or the hyperlink takes you to Account Financial History - Main
Payoff Balance	The account's payoff balance. It will only be displayed if the amount differs from the current balance.	Not applicable
Last Bill	The date and amount of the account's last bill along with its due date.	Bill - Main
Last Payment	The date and amount of the account's last payment.	Payment Event - Main
Previous Bill	The date and amount of the bill prior to the last bill for the account.	Bill - Main
Next Bill Date	The next date on which a bill is scheduled to be produced for the account (based on the account's bill cycle's schedule)	Not applicable
Pending Bill Exists	The date of the pending bill. This row only appears if there is a pending bill associated with the account	Bill - Main
Freezable Bill Segments	If `freezable` bill segments exist, this row contains the number of segments and their total amount.	Bill Segment - Main
Incomplete Adjustments	If `incomplete` adjustments exist, this row contains the number of adjustments and their total amount.	Adjustment - Main

Copyright © 2024, Oracle and/or its affiliates.

Label	What Is Displayed	Drill Down Transfers You To
Freezable Adjustments	If `freezable` adjustments exist, this row contains the number of adjustments and their total amount.	Adjustment - Main
Payment(s) Pending Upload	Appears when the account has a `pending` payment staging record (i.e., the account has an external payment that has not been uploaded into the system). To prevent payments from being created and frozen before the actual payment is received, external payments are not uploaded into the system until their accounting date is reached. Note that the amount displayed represents the amount tendered for the account, but does not reflect how the payment may be distributed (i.e., the payment may be distributed to other accounts). For information about payment staging records, refer to Interfacing Payments from External Sources.	Payment Upload Staging
Incomplete Payment(s)	Appears when the account has payments that are `incomplete`. This row contains the number of payments and their total amount. Note, `incomplete` automatic payments that are pending distribution are not included in this alert as they are highlighted on the **Auto Pay will be distributed on** described below.	Payment - Main
Payment(s) with Errors	Appears when the account has payments that are in `error`. This row contains the number of payments and their total amount.	Payment - Main
Freezable Payments)	If `freezable` payments, this row contains the number of payments and their total amount.	Payment Event - Main
Auto Pay will be created on	This row shows the date that the next auto pay will be created. This alert will appear after a bill is completed for an account on auto pay and when your installation option's **Auto Pay Creation Option** is `Create on Extract Date`. If the installation option is `Create at Bill Completion`, then the Autopay will be created immediately at bill completion time and no information is displayed here. Refer to How And When Are Automatic Payments Created? for more information.	Bill - Main
Auto Pay will be distributed on	This row shows the date that the next auto pay will be distributed. This alert will appear after a bill is completed for an account on auto pay and when your installation option's **Auto Pay Creation Option** is `Create on Extract Date`. If the installation option is `Create at Bill Completion`, then the payment for the Autopay will be created immediately at bill completion time and no information is displayed here. Refer to How And When Are Automatic Payments Created? for more information.	Payment Event - Main

Copyright © 2024, Oracle and/or its affiliates.

Alert Zone

The **Alert Zone** is a grid that contains messages highlighting a variety of situations. Clicking on the hyperlink transfers you to an appropriate page. The following table lists the various alerts that may appear and the page to which you will be transferred if click on the hyperlink. The table lists the alerts in alphabetical order, but it includes a column indicating the order in which the alert will appear.

Note: You can create additional alert conditions. The following table contains those alerts supported in the base package. It is possible to highlight additional conditions by plugging-in the appropriate algorithm(s) on the installation record. Refer to Count Customer Contacts of a Given Type and Count Promise To Pay with a Given Status for examples of such algorithms.

Note: Account alerts are implementation-specific. On Account - Alerts, a user can define account-specific alerts that should be displayed whenever the account is selected. The Alert Types control table contains the possible alert messages.

The following table describes the different alert messages that appear in the **Alerts** zone:

Alert Text	Order	Description	Drill Down Transfers You To
Account alerts	6	Appears for each user-defined alert associated with an account.	Account - Alert
Active Write Off Process	10	Appears when an account has an `active` write off process. Refer to How Are Write off Processes Cancelled for information describing when a write-off process is deactivated.	Write Off Process
Auto-Pay Active: ***auto pay method description***	25	Appears when the account has an automatic payment option effective on the current date. The auto pay method description (i.e. `Direct Debit` or `Payment Advice`) appears only if payment advice functionality is enabled. Refer to Payment Advices for more information on payment advice functionality.	Account - Auto Pay
Comment Exists On Account	5	Appears when a freeform comment has been entered for the account (on Account - Main)	Account
Held Order Exists	32	Appears when the person has a `held` order.	Order
Installation dynamic alerts	42	You can install plug-in algorithms on the installation record that will create alerts (if given situations exist). Refer to Count Customer Contacts of a Given Type and Count Promise To Pay with a Given Status for examples of such algorithms	Varies by plug-in
Last Contact: ***days old - user name who added the contact***	3	Appears when the person has a customer contact. The "days old" presents how old the contact is (based on the contact's date). The word `Today` is shown if the last contact was on the current date. The word `Yesterday` is shown if the last contact was on the current date - 1 day. If neither is applicable, the number of days old is shown.	Customer Contact

Copyright © 2024, Oracle and/or its affiliates.

Alert Text	Order	Description	Drill Down Transfers You To
Non-cash Deposit	15	Appears when the account has non-cash deposits effective on the current date	Account - Deposit
n Open Contact(s) for Person	4	Appears if there are `open` customer contacts associated with the person. Refer to Customer Contacts Can Be Used As Case Files for more information.	Customer Contact
Pending Order Exists	31	Appears when the person has a `pending` order.	Order
Pending Proposal Contract Exists	36	Appears when the account has a proposal contract in the `pending` state. Refer to *Proposal Contract State Transition* for more information.	Contract
Pending Quote Exists	34	Appears when the account has a `pending` quote. Refer to Quote Lifecycle for more information.	Quote
Pending Stop	18	Appears when there is a contract in the state of `pending stop` linked to the account. Refer to The Lifecycle Of A Pending Stop Contract for more information.	Start/Stop
n Persons on Account	26c	Appears when the account has multiple persons linked to it. An account's persons are maintained on the Account - Person page.	Account - Person
Quotable Proposal Contract Exists	35	Appears when the account has a proposal contract in the `quotable` state. Refer to *Proposal Contract State Transition* for more information.	Contract
Quote Awaiting Customer Response Exists	33	Appears when the account has a `complete` quote with a quote detail that references a `quotable` proposal contract. Refer to Quote Lifecycle for more information.	Quote
Reactivated Contracts Exist	21	Appears when any of the account's contracts exist in the state of `reactivated` (i.e., a financial transaction has been generated after the contract was `closed`). Refer to The Lifecycle Of A Contract for more information.	Contract
Contract type alert message	13	Appears when the customer has a contract with an contract type that has been marked with an alert message (refer to Contract Type - Details) Note that if the account in control central has more than one contract, pressing the drill down button will also launch the contract search dialog.	Contract

Copyright © 2024, Oracle and/or its affiliates.

Alert Text	Order	Description	Drill Down Transfers You To
Seasonal Address Exists	38b	Appears if the account has a seasonal address and there is no seasonal address active on the current date.	Person - Contact Information
Seasonal Address Currently Effective (MM/DD - MM/DD)	38a	Appears if the account has an active seasonal address that is effective on the current date. The date range that the seasonal address is effective is included in the alert.	Person - Contact Information
Statement Construct Active	28	Appears if the person has an `active` statement construct	Statement Construct

Setting Up Bill Print Groups

Bill print groups allow you to categorize an account's contracts into groups for bill *print* purposes.

Note: Bill print groups are optional. Typically, only accounts with many contracts will use bill print groups because the standard bill print priority is sufficient for accounts with a limited number of contracts. Refer to Contract Type - Billing for more information about the standard bill print priority.

Let's use an example to clarify the bill print group concept. Consider a local government's account. This account would have many contracts (some for the police department, others for the court system, others for the department of public works, etc.).

If you don't create a bill print group for this customer, their bill segments will be printed in the order dictated by each segment's contract's contract type's bill print priority. If you create a bill print group for this customer, you can define the customer's desired categories (e.g., police, courts, DPW, etc.) and then link each of the account's contracts to the appropriate category.

Note: Nomenclature. We refer to the categories under a bill print group as "subgroups". A bill print group can have an unlimited number of subgroups. When you create a subgroup, you define its relative bill print priority and the associated verbiage to print on the bill (assuming you print information about the subgroup on the bill).

We'd like to highlight the following characteristics of bill print groups:

- A bill print group is associated with a specific account. If multiple customers have the exact categorization preferences, you will have to set up multiple bill print groups.
- Over time, a customer could have many bill print groups. This would happen if a customer's bill categorization preferences change over time. We'd like to stress that while an account can have multiple bill print groups, only one will be used by the bill print process (the one effective on the bill date).
- Bill print groups only affect printed bills. Bill print groups do not affect the order in which contracts appear on the bill maintenance transaction.
- If new contracts are added after a bill print group is set up for an account, they will not be linked to a subgroup. If you neglect to link the new contracts to one of the bill print group's subgroups they will be printed under the "default" subgroup (every bill print group must have one default subgroup to cater for this situation).

CAUTION: Important! While the system supports the definition of bill print groups and the categorization of an account's contracts into the various subgroups, the base package's bill print extract does NOT take advantage of this information.

The topics in this section describe how to set up a bill print group and how to link an account's contracts to its subgroups.

Copyright © 2024, Oracle and/or its affiliates.

Bill Print Group - Main

Open **Billing Management, Bill Print Group** to maintain an account's bill print group and subgroups. After defining this information, transfer to the **Contract Sub Group** tab to link the account's contracts to the subgroups.

Description of Page

The **Bill Print Group ID** is displayed on every page. This value only appears after the bill print group exists on the database. The ID is a system-assigned random number that stays with a bill print group for life.

Enter the **Account ID** associated with the bill print group.

Use **Effective Date** to define when the bill print group is effective for the account. This date is important as it allows a customer to change their preferences over time. For example, if the customer wants to change the number of subgroups on a given date, you would simply add a new bill print group effective on this date and then define the new subgroups (and link each contract to one of the subgroups).

Use **Status** to define if the bill print group is `Active` or `Inactive`. You would only use the `Inactive` value if the bill print group is no longer needed (as there is no delete action on this transaction).

Enter a brief **Description** of the bill print group.

Use **Comments** to describe anything special about the bill print group.

The grid contains the bill print group's subgroups. The following information should be defined for each subgroup:

- **Bill Print Sub Group.** This is the unique identifier of the bill print subgroup.
- **Sub Group Bill Print Priority.** This is the relative print priority of the subgroup in respect of the other subgroups.
- **Use as Default.** Turn on this switch for the default sub group. A bill print group must have one and only one default subgroup. The default group is used by the bill print process if it detects a contract that is not linked to a subgroup (it links this contract to the default subgroup).
- **Sub Group Description.** This is a brief description of the subgroup.
- **Description on Bill.** This is the verbiage that will print on the bill (assuming you print something on your bills for the subgroup).

After defining the bill print group's subgroups, navigate to the **Contract Sub Group** tab to link the account's contracts to the subgroups.

Bill Print Group - Contract Sub Group

This page is used to link the account's contracts to one of the bill print group's subgroups.

Open **Billing Management, Bill Print Group** and navigate to the **Contract Sub Group** page to maintain this information.

Description of Page

The **Bill Print Group's ID**, **Account ID** and **Effective Date** are displayed at the top of the page.

The filters control the contracts that appear in the **Contracts in Bill Print Group** grid. The following points describe the various options:

- Use the **Contract Filter** to define the types of contracts that appear in the grid. The following options are available:
  - `Address.` Use this option to restrict contracts to those linked to service points associated with a given **Address**, **City** and/or **Postal** code. Note, you can specify any combination of these fields.
  - `All.` Use this option if you do not wish to restrict contracts based on contract attributes.
  - `Bill Print Sub Group.` Use this option to restrict contracts to those that are linked to a given **Bill Print Sub Group**.
  - `Geographic Type.` Use this option to restrict contracts to those linked to service points associated with a given **Geo Type** and **Value**.

Copyright © 2024, Oracle and/or its affiliates.

  - `Contract Type.` Use this option to restrict contracts to those linked to a given **Division** and **Contract Type**.
- Use **Status Filter** to restrict the contracts based on their status. The following options are available:
  - `All`. This option shows all contracts regardless of status.
  - Refer to Contract Lifecycle for the various status values.

Don't forget to click the search button after changing the filters.

The **Contracts in Bill Print Group** contains an entry for every non-cancelled contract linked to the account that is linked to one of the bill print group's subgroups. The following information appears in the grid:

- Use **Bill Print Sub Group** to define the subgroup associated with the contract.
- Use **Sequence** when there is more than one contract in the subgroup. The sequence controls the order in which the contract's financial information appears on the bill.
- The **Contract Information** column provides a brief description of the contract.
- The **contract** column contains the unique identifier of the contract.
- The **Location Information** column contains the characteristic location associated with the bill segment's contract.

The next set of filters control the contracts that appear in the **Contracts Not in Bill Print Group** grid. The following points describe the various options:

- Use the **Contract Filter** to define the types of contracts that appear in the grid. The following options are available:
  - `Address.` Use this option to restrict contracts to those linked to service points associated with a given **Address**, **City** and/or **Postal** code. Note, you can specify any combination of these fields.
  - `All.` Use this option if you do not wish to restrict contracts based on contract attributes.
  - `Geographic Type.` Use this option to restrict contracts to those linked to service points associated with a given **Geo Type** and **Value**.
  - `Contract Type.` Use this option to restrict contracts to those linked to a given **Division** and **Contract Type**.
- Use **Status Filter** to restrict the contracts based on their status. The following options are available:
  - `All`. This option shows all contracts regardless of status.
  - Refer to Contract Lifecycle for the various status values.

Don't forget to click the search button after changing the filters.

The **Contracts Not in Bill Print Group** contains an entry for every non-cancelled contract linked to the account that is NOT linked to one of the bill print group's subgroups. The following information appears in the grid:

- Use **Bill Print Sub Group** to define the subgroup associated with the contract.
- Use **Sequence** when there is more than one contract in the subgroup. The sequence controls the order in which the contract's financial information appears on the bill.
- The **Contract Information** column provides a brief description of the contract.
- The **Contract** column contains the unique identifier of the contract.
- The **Location Information** column contains the characteristic location associated with the bill segment's contract.

Address

Oracle Revenue Management and Billing enables you to define various types of address for a person, such as:

Type of Address	Purpose
Person Mailing	This address is used to sent the bills, letters, quotes, and statements to a person.
Person Seasonal	This address is only used when the person wants to send their correspondence to an alternate address for a predefined period.
Account Override	This address is used when the person wants the account-specific correspondence to be sent to an override address instead of the person's mailing or seasonal address.

Copyright © 2024, Oracle and/or its affiliates.

The system provides the ability to define effective dated address for various entities, such as person, account, or adjustment. For example, for a person named P1, you can define two mailing addresses (A1 and A2) - A1 is effective from Jan 2017 and A2 is effective from Jan 2018. Accordingly, all correspondence for P1 from Jan 2017 to Dec 2017 will be sent to the address A1 and all correspondence for P1 from Jan 2018 onwards will be sent to the address A2. A new screen named **Address** is introduced to define effective dated address. You can also edit or delete an effective dated address from the **Address** screen.

Note: If the **Enable Effective Dated Address Feature** option type of the **C1-Address** feature configuration is set to **N**, you cannot edit the entity details or disassociate an entity from an address from the **Address** screen when the address is associated to the entity using the **Person Mailing**, **Person Seasonal**, or **Account Override** address type.

You can optionally configure the system to use the approval workflow process whenever an address is defined, edited, or deleted from the **Address** screen. If the approval workflow process is configured for the **C1-Address** business object, the approval transaction is created while defining, editing, copying, or deleting an address. Once the approver approves the approval transaction, the changes are reflected in the system. While configuring the approval workflow group for the **C1-Address** business object, you need to use the **C1-AddressAppTxnDisplay** display UI map and **C1-AddressAppTxnInput** input UI map.

Note:

Now, the following screens or tabs use a new business service named **C1-GetEffectiveDatedAddress**:

- **Contact Information** and **Address** tabs on the **Person** screen
- **Persons** tab on the **Account** screen
- **A/P Request** tab on the **Adjustment** screen
- **Overdue Process Information for Account: {Account Info}** screen (accessed through the **Delinquency Central** screen)
- **Main** tab on the **Refund Request** screen
- **Bill Routings** tab on the **Bill** screen
- **Main** tab on the **Customer Contact** screen
- **Quote Routings** tab on the **Quote** screen
- **Main** tab on the **Statement** screen
- **Main** tab on the **Order** screen

This business service is also used during the statement generation, bill extract, and while displaying effective seasonal address alert on the dashboard. For more information about this business service, see C1-GetEffectiveDatedAddress.

Prerequisites

To setup the effective dated address feature, you need to do the following:

- Define the required countries and their states in the system.
- Define values for the **HOUSE_TYPE** and **ADDRESS_TYPE_FLG** lookup fields.
- Define the required characteristic types where the characteristic entity is set to **Address**.
- Set the **Enable Effective Dated Address Feature** option type of the **C1-Address** feature configuration.

Related Topics

For more information on...	See...
How to setup the **C1-Address** feature configuration	Setting the C1-ADDRESS Feature Configuration

Copyright © 2024, Oracle and/or its affiliates.

Maintaining Address Status through Inbound Web Service

Until now, whenever you updated the address details of an entity (i.e. person or account) through a health care inbound message or through the membership repricing inbound web service, the system used to override the existing address of the entity. However, the business needs to maintain the old address history of an entity for the audit purpose. Now, the system enables you to maintain the status of an address for an entity through a health care inbound message and through the membership repricing inbound web service. This helps you to maintain both active and inactive addresses of the entity in the system. To implement this feature, the following changes are made in the system:

- A new field named **BO_STATUS_CD** is added in the **C1_ADDRESS** table.
- A new lookup named **ADDR_STATUS_FLG** is added in the system.
- A new tag named **addressStatus** is introduced in the **C1-Address** and **C1-HCInboundMessage** business object schema.
- A new tag named **addressStatus** is introduced in the **C1-MembershipRepricing** business service schema.
- The **C1-HCINPROC** algorithm type is enhanced to maintain both active and inactive addresses of the entity in the system.

Whenever you create an address for an entity through a health care inbound message or through the membership repricing inbound web service, you can specify the status in the **addressStatus** tag. Note that while creating an address for an entity, the status of the address should always be set to **C1AC** (i.e. Active). The **addressStatus** tag is not mandatory. Therefore, if you do not specify the value for the **addressStatus** tag while creating an address, the system, by default, sets it to **C1AC** (i.e. Active). Whenever you create an address for an entity from the user interface (i.e. through the **Address** screen), you cannot set the status of the address. By default, it is set to **Active**.

You can change the address status of an entity only through a health care inbound message or through the membership repricing inbound web service and not from the user interface (i.e. through the **Address** screen). While updating an address or while changing the status of an address through a health care inbound message or through the membership repricing inbound web service, you need to specify the details such as entity type, entity ID, effective date, and address type to identify the address whose details you want to change. You can change the status of an address from **C1AC** (i.e. Active) to **C1IN** (i.e. Inactive) and vice-versa.

While changing the status of an address from **C1AC** (i.e. Active) to **C1IN** (i.e. Inactive), the system checks whether an active address for the entity type, entity ID, effective date, and address type combination already exists in the system. If the active address is not found for the given combination, an appropriate error message is shown while processing the health care inbound message or the membership repricing inbound web service. However, if the active address is found for the given combination, the system checks whether the active address is referred in any trial bill routing, bill routing, or statement construct. If the active address is referred in the system, the status of the address is changed to **C1IN** (i.e. Inactive). However, if the active address is not referred in the system, the address of the entity is deleted from the system.

While receiving an address for an entity with the **C1AC** (i.e. Active) status, the system checks whether an active address for the entity type, entity ID, effective date, and address type combination already exists in the system. If the active address is found for the given combination, the system updates the existing address record of the entity. However, if the active address is not found for the given combination, the system checks whether an inactive address for the entity type, entity ID, effective date, and address type combination already exists in the system. If the inactive address is found for the given combination, the system changes the status of the address to **C1AC** (i.e. Active). However, if the inactive address is not found for the given combination, the system creates the address for the entity type, entity ID, effective date, and address type combination with the **C1AC** (i.e. Active) status. Note that, at a time, only one active address can exist for the entity type, entity ID, effective date, and address type combination in the system.

Whenever you create or update an address or change the status of an address through a health care inbound message or through the membership repricing inbound web service, the system creates a log entry for the address. You can view the logs of an address in the **Log** tab of the **Address Information** screen.

Copyright © 2024, Oracle and/or its affiliates.

Note:

- At present, this feature is offered only for the health insurance domain and not for the financial services domain.
- While upgrading ORMB Version from 5.1.0.0.0 or 6.0.0.0.0 to 6.1.0.0.0, the system will automatically set the status of the existing address records of an entity to **Active**. Note that customers from the financial services domain cannot change the status of the existing address records of an entity to **Inactive**.

Address

The **Address** screen allows you to search for an address using various search criteria. It also allows you to define an address. It contains the following zones:

- Search Address on page 111
- Address Characteristics on page 116
- Address Entities on page 115

Through this screen, you can navigate to the following screen:

- Address Information on page 123

Search Address

The **Search Address** zone allows you to search for an address using various search criteria. It appears in the **Address** screen as well as in the **Person - Address** tab. It contains the following two sections:

- **Search Criteria** - The **Search Criteria** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Search By	Indicates that you want to search for an address using the address details. At present, there is only one valid value - **Address Details**. **Note:** By default, the **Address Details** option is selected.	Not applicable
Address ID	Used to search a particular address.	No
Person Name	Used to search addresses which are associated to one or more persons with the same name.	No
Country	Used to search addresses which belong to a particular country.	No
State	Used to search addresses which belong to a particular state.	No
Address Line 1	Used to search addresses with a particular address line 1.	No
City	Used to search addresses which belong to a particular city.	No
Postal	Used to search addresses with a particular postal code.	No
County	Used to search addresses which belong to a particular county.	No
Address Line 2	Used to search addresses with a particular address line 2. **Note:** This field appears when you click the **Advanced Search** link available in the right side of the **Search Address** zone.	No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	.	
Address Line 3	Used to search addresses with a particular address line 3. **Note:** This field appears when you click the **Advanced Search** link available in the right side of the **Search Address** zone.	No
Address Line 4	Used to search addresses with a particular address line 4. **Note:** This field appears when you click the **Advanced Search** link available in the right side of the **Search Address** zone.	No
Geographic Code	Used to search addresses which belong to a particular geographic code. **Note:** This field appears when you click the **Advanced Search** link available in the right side of the **Search Address** zone.	No
Entity Type	Used to search addresses which are associated to a particular type of entity. The valid values are: • Account • Adjustment • Person **Note:** This field appears when you click the **Advanced Search** link available in the right side of the **Search Address** zone. By default, the entity type is set to **Person** when you are searching the address using the person name as the search criteria. By default, the entity type is set to **Person** when you are viewing the address of the person through the **Address** tab in the **Person** screen.	No
Entity ID	Used to search addresses which are associated to a particular entity.	No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: This field appears when you click the **Advanced Search** link available in the right side of the **Search Address** zone. This field is enabled only when you set the entity type to **Account**, **Person**, or **Adjustment**. This field is disabled when you are searching the address using the person name as the search criteria. The **Search** (🔍) icon appears corresponding to the **Entity ID** field. On clicking the **Search** icon, the **Account Search**, **Person Search**, or **Adjustment Search** window appears depending on the type of the entity that you have selected. By default, the ID of the person whose address details you are viewing through the **Address** tab of the **Person** screen appears in the **Entity ID** field.	
Address Type	Used to search addresses with a particular address type. The valid values are: • Account Override • Person Mailing • Person Seasonal **Note:** This field appears when you click the **Advanced Search** link available in the right side of the **Search Address** zone.	No
Effective Date	Used to search addresses which are effective for an entity on a particular date. **Note:** This field appears when you click the **Advanced Search** link available in the right side of the **Search Address** zone.	No
Season Starting Month Day (MM-DD)	Used to search seasonal addresses which are effective from a particular date. **Note:** This field appears when you click the **Advanced Search** link available in the right side of the **Search Address** zone and select the **Person Seasonal** option from the **Address Type** list.	No
Season Ending Month Day (MM-DD)	Used to search seasonal addresses which are effective till a particular date. **Note:** This field appears when you click the **Advanced Search** link available in the right side of the **Search Address** zone and select the **Person Seasonal** option from the **Address Type** list.	No
Status	Used to search addresses with a particular status. The valid values are:	No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	• Active • Inactive	
	Note: This field appears when you click the **Advanced Search** link available in the right side of the **Search Address** zone.	

Note: You must specify at least one search criterion while searching for an address. One more search criterion (such as, **Address ID**, **Person Name**, **Country**, **State**, **County**, **Entity Type**, **Entity ID**, or **Address Type**) is required when you are searching for an address using the **Address Line 2**, **Address Line 3**, **Address Line 4**, or **Effective Date** field.

- **Search Results** - On clicking the **Search** button, the search results appear based on the specified search criteria. The **Search Results** section contains the following columns:

Column Name	Column Description
Address Information	Displays information about the address.
	Note: It has a link. On clicking the link, the **Address Information** screen appears where you can view the details of the respective address.
Address Line 1	Displays the first line of the address. It may contain details, such as house number and apartment name.
Address Line 2	Displays the second line of the address. It may contain details, such as street name.
Address Line 3	Displays the third line of the address. It may contain the landmark details.
Address Line 4	Displays the fourth line of the address.
Geographic Code	Displays the geographic code of the address.
Country	Displays the country to which the address belongs.
City	Displays the city to which the address belongs.
State	Indicates the state where the city or county is located.
Postal	Displays the postal or zip code of the address.
County	Displays the county to which the address belongs.
Season Starting Month Day (MM-DD)	Displays the date (in the MM-DD format) from when the seasonal address is effective.
Season Ending Month Day (MM-DD)	Displays the date (in the MM-DD format) till when the seasonal address is effective.
Status	Indicates the status of the address. The valid values are: • Active • Inactive

Note: Pagination is used to display limited number of records in the **Search Results** section. You can use the navigation links, such as **Previous** and **Next** to navigate between pages.

Copyright © 2024, Oracle and/or its affiliates.

On clicking the **Broadcast** (📶) icon corresponding to an address, the **Address Entities** and **Address Characteristics** zones appear.

Note: The **Address Entities** zone appears only when the address is associated to an entity. Similarly, the **Address Characteristics** zone appears when a characteristic is defined for the address.

Related Topics

For more information on...	See...
How to search for an address	Searching for an Address on page 116
How to view the details of an address	Viewing the Address Details on page 117
How to view the entities associated to an address	Viewing the Entities Associated to an Address on page 117
How to view the characteristics of an address	Viewing the Characteristics of an Address on page 118

Address Entities

The **Address Entities** zone lists the entities to which the address is associated. It appears in the **Address** screen as well as in the **Person - Address** tab. However, it appears when the address is associated to an entity. It contains the following columns:

Column Name	Column Description
Effective Date	Displays the date from when the address is associated to the entity.
Address Type	Indicates the type of the address. The valid values are: • Account Override • Person Mailing • Person Seasonal
Entity Type	Indicates the type of the entity to which the address is associated. The valid values are: • Account • Adjustment • Person
Entity ID	Displays the entity ID.
Entity Information	Displays information about the entity. In addition, this column has a context menu which helps in navigating to other screens in the application. **Note:** It has a link. On clicking the link, the **Person**, **Account**, or **Adjustment** screen appears where you can view the details of the respective entity.

By default, the **Address Entities** zone does not appear in the **Address** screen. It appears when you click the **Broadcast** (📶) icon corresponding to the address in the **Search Address** zone.

You can change the layout of this zone. For more information on how to change the layout, see Changing the Layout.

Related Topics

For more information on...	See...
How to view the entities associated to an address	Viewing the Entities Associated to an Address on page 117

Copyright © 2024, Oracle and/or its affiliates.

Address Characteristics

The **Address Characteristics** zone lists the characteristics of the address. It appears in the **Address** screen as well as in the **Person - Address** tab. However, it appears when a characteristic is defined for the address. It contains the following columns:

Column Name	Column Description
Effective Date	Displays the date from when the characteristic is effective for the address.
Characteristic Type	Indicates the characteristic type.
Characteristic Value	Displays the value of the characteristic type.

By default, the **Address Characteristics** zone does not appear in the **Address** screen. It appears when you click the **Broadcast** (📶) icon corresponding to the address in the **Search Address** zone.

You can change the layout of this zone. For more information on how to change the layout, see Changing the Layout.

Related Topics

For more information on...	See...
How to view the characteristics of an address	Viewing the Characteristics of an Address on page 118

Searching for an Address

Prerequisites

To search for an address, you should have:

- Countries and their states defined in the application
- Address types defined in the **ADDRESS_TYPE_FLG** lookup field

Procedure

To search for an address:

1. Click the **Menu** link in the **Application** toolbar.
   A list appears.
2. From the **Main** menu, select **Customer Management** and then click **Address**.
   A sub-menu appears.
3. Click the **Search** option from the **Address** sub-menu.
   The **Address** screen appears.
4. Enter the required search criteria in the **Search Address** zone.

   Note:

   The **Advanced Search** link appears in the right side of the **Search Address** zone. On clicking the **Advanced Search** link, additional fields appear in the zone which help you to refine the search.

   ORMB search engine supports wildcard search, where you can substitute the percentage (%) symbol as a stand in for any word or letter in a search criteria. You can use the '%' wildcard character in all input fields except the date and ID fields. The '%' wildcard character is suffixed automatically at the end of the partial search criteria. Therefore, you may or may not specify the wildcard character at the end of the partial search criteria. However, you have to prefix the wildcard character manually wherever required.

5. Click **Search**.
   A list of addresses that meets the search criteria appears in the **Search Results** section.

Related Topics

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
Address screen	Address on page 111
Search Address zone	Search Address on page 111

Viewing the Address Details

Procedure

To view the details of an address:

1. Search for the address in the **Address** screen.
2. In the **Search Results** section, click the link in the **Address Information** column corresponding to the address whose details you want to view.
   The **Address Information** screen appears.
3. Ensure that the **Main** tab is selected.
4. View the details of the address in the **Address** zone.

Related Topics

For more information on...	See...
How to search for an address	Searching for an Address on page 116
Address Information screen	Address Information on page 123
Address zone	Address on page 123

Viewing the Entities Associated to an Address

Procedure

To view the entities associated to an address:

1. Search for the address in the **Address** screen.
2. In the **Search Results** section, click the **Broadcast** (📶) icon corresponding to the address whose details you want to view.
   The **Address Entities** and **Address Characteristics** zones appear.

   Note: The **Address Entities** zone appears when the address is associated to an entity. Similarly, the **Address Characteristics** zone appears when a characteristic is defined for the address.

3. View the entities to which the address is associated in the **Address Entities** zone.

Related Topics

For more information on...	See...
How to search for an address	Searching for an Address on page 116
Address Entities zone	Address Entities on page 115
Address Characteristics zone	Address Characteristics on page 116

Copyright © 2024, Oracle and/or its affiliates.

Viewing the Characteristics of an Address

Procedure

To view the characteristics of an address:

1. Search for the address in the **Address** screen.
2. In the **Search Results** section, click the **Broadcast** (📶) icon corresponding to the address whose details you want to view.
   The **Address Entities** and **Address Characteristics** zones appear.

   Note: The **Address Entities** zone appears when the address is associated to an entity. Similarly, the **Address Characteristics** zone appears when a characteristic is defined for the address.
3. View the characteristics of the address in the **Address Characteristics** zone.

Related Topics

For more information on...	See...
How to search for an address	Searching for an Address on page 116
Address Entities zone	Address Entities on page 115
Address Characteristics zone	Address Characteristics on page 116

Defining an Address

Prerequisites

To define an address, you should have:

- Countries and their states defined in the application
- House types defined in the **HOUSE_TYPE** lookup field

Procedure

To define an address:

1. Click the **Menu** link in the **Application** toolbar.
   A list appears.
2. From the **Main** menu, select **Customer Management** and then click **Address**.
   A sub-menu appears.
3. Click the **Add** option from the **Address** sub-menu.

   The **Address** screen appears. It contains the following sections:

   - **Main** - Used to specify basic details about the address.
   - **Characteristics** - Used to define the characteristics for the address.
   - **Address Entities** - Used to associate the address with one or more entities, such as person, account, or adjustment.

   The **Main** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Country	Used to indicate the country.	Yes

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: The list includes only those countries which are already defined in the system. On selecting a country, only those address fields which are set as **Optional** while defining the country appear in the **Main** section. Also, the labels of the address fields will change depending on the business label which is defined for the respective address field in the **Country** screen.	
State	Used to indicate the state where the city or county is located. **Note:** The states listed in this field change depending on the country that you have selected.	No
Address 1	Used to specify the first line of the address. It may contain details, such as house number and apartment name.	No
Address 2	Used to specify the second line of the address. It may contain details, such as street name.	No
Address 3	Used to specify the third line of the address. It may contain the landmark details.	No
Address 4	Used to specify the fourth line of the address.	No
City	Used to specify the city name.	No
Number 1	Used to specify the numeric information, if any, related to the address.	No
Number 2	Used to specify the numeric information, if any, related to the address.	No
County	Used to specify the county name.	No
Postal	Used to specify the postal or zip code of the address.	No
House Type	Used to indicate the type of the house. **Note:** The list includes only those values which are defined in the **HOUSE_TYPE** lookup field.	No
In City Limit	Used to indicate whether the address is within the city limit. The valid values are: • Yes • No	No
Geographic Code	Used to specify the geographic code of the address.	No

Tip: Alternatively, you can access this screen by clicking the **Add** button in the **Page Title** area of the **Address** screen.

4. Enter the required details in the **Main** section.

Copyright © 2024, Oracle and/or its affiliates.

5. Define characteristics for the address, if required.
6. Associate address with one or more entities, if required.
7. Click **Save**.
   The address is defined and the address ID is generated.

> **Note:** If the approval workflow is configured for the **C1-Address** business object, the approval transaction is created on clicking the **Save** button. Once the approver approves the approval transaction, the address is defined and the address ID is generated.

Related Topics

For more information on...	See...
How to define a characteristic for an address	Defining a Characteristic for an Address on page 120
How to associate an address to an entity	Associating an Address to an Entity on page 121

Defining a Characteristic for an Address

Prerequisites

To define a characteristic for an address, you should have:

- Characteristic types defined in the application (where the characteristic entity is set to **Address**)

Procedure

To define a characteristic for an address:

1. Ensure that the **Characteristics** section is expanded when you are defining, editing, or copying an address.

   The **Characteristics** section contains the following fields in a grid:

Field Name	Field Description	Mandatory (Yes or No)
Effective Date	Used to specify the date from when the characteristic is effective for the address.	Yes (Conditional) **Note:** This field is required when you are defining a characteristic for the address.
Characteristic Type	Used to indicate the characteristic type. **Note:** The list includes only those characteristic types where the characteristic entity is set to **Address**.	Yes (Conditional) **Note:** This field is required when you are defining a characteristic for the address.
Characteristic Value	Used to specify the value for the characteristic type.	Yes (Conditional)

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: If you select a predefined characteristic type, the **Search** (🔍) icon appears corresponding to the **Characteristic Value** field. On clicking the **Search** icon, the **Predefined Characteristic Search** window appears. On specifying the value for a predefined characteristic type, the description of the characteristic value appears corresponding to the **Characteristic Value** field.	**Note:** This field is required when you are defining a characteristic for the address.

2. Enter the required details in the **Characteristics** section.
3. If you want to define more than one characteristic for the address, click the **Add** (+) icon and then repeat step 2.

> **Note:** However, if you want to remove a characteristic from the address, click the **Delete** (🗑) icon corresponding to the characteristic.

4. Click **Save**.
   The characteristics are defined for the address.

Related Topics

For more information on...	See...
How to define an address	Defining an Address on page 118
How to edit an address	Editing an Address on page 125
How to copy an address	Copying an Address on page 127

Associating an Address to an Entity

Prerequisites

To associate an address to an entity, you should have:

- Required entity defined in the application
- Address types defined in the **ADDRESS_TYPE_FLG** lookup field

Procedure

To associate an address to an entity:

1. Ensure that the **Address Entities** section is expanded when you are defining, editing, or copying an address.

   The **Address Entities** section contains the following fields in a grid:

Field Name	Field Description	Mandatory (Yes or No)
Effective Date	Used to specify the date from when the address is associated with the entity.	Yes (Conditional) **Note:** This field is required when you are associating the address to an entity.

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
Entity Type	Used to indicate the type of the entity. The valid values are: • Account • Adjustment • Person	Yes (Conditional) **Note:** This field is required when you are associating the address to an entity.
Entity ID	Used to indicate the entity to which you want to associate the address. **Note:** By default, this field is disabled. If you select the entity type, the **Search** (🔍) icon appears corresponding to the **Entity ID** field. On clicking the **Search** icon, the **Account Search**, **Person Search**, or **Adjustment Search** window appears depending on the entity type that you have selected.	Yes (Conditional) **Note:** This field is required when you are associating the address to an entity.
Address Type	Used to indicate the type of the address. The valid values are: • Account Override • Person Mailing • Person Seasonal **Note:** The list includes only those values which are already defined in the **ADDRESS_TYPE_FLG** lookup field. You can only set the address type to **Person Mailing** or **Person Seasonal** when the entity type is set to **Person**. And, you can only set the address type to **Account Override** when the entity type is set to **Account**.	Yes (Conditional) **Note:** This field is required when you are associating the address to an entity.
Account-Person ID	Used to indicate the account holder to whom you want to associate the address. **Note:** This field is enabled when the address type is set to **Account Override**. The **Search** (🔍) icon appears corresponding to the **Account-Person ID** field. On clicking the **Search** icon, the **Account Person Search** window appears.	Yes

2. Enter the required details in the **Address Entities** section.
3. If you want to associate more than one entity with the address, click the **Add** (+) icon and then repeat step 2.

> **Note:** However, if you want to dissociate an entity from the address, click the **Delete** (🗑) icon corresponding to the entity.

4. Click **Save**.
   The address is associated with the entities.

> **Note:** You cannot associate the address to an entity if the entity is already associated to another address using the same address type and effective date.

Related Topics

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
How to define an address	Defining an Address on page 118
How to edit an address	Editing an Address on page 125
How to copy an address	Copying an Address on page 127

Address Information

The **Address Information** screen allows you to:

- View the details of the address
- Edit the details of the address
- Copy the address
- Delete the address
- View the log of the address
- Add a log entry for the address

This screen consists of the following tabs:

- Address Information - Main on page 123
- Address Information - Log on page 130

Address Information - Main

The **Main** tab displays the basic details and characteristics of an address and the entities to which an address is associated. It contains the following zone:

- Address on page 123

Address

The **Address** zone displays the details of the address. It contains the following sections:

- **Main** - Displays the basic information about the address. It contains the following fields:

Field Name	Field Description
Address Information	Displays information about the address.
Address ID	Displays the address ID.
Country	Indicates the country to which the address belongs.
	Note: The remaining fields in this section appear when the respective data is available in the address. Also, the labels of the address fields will change depending on the business label defined for the respective address field in the **Country** screen.
City	Displays the city name.
Address 1	Displays the first line of the address. It may contain details, such as house number and apartment name.
Address 2	Displays the second line of the address. It may contain details, such as street name.
Address 3	Displays the third line of the address. It may contain the landmark details.

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description
Address 4	Displays the fourth line of the address.
Number 1	Displays the numeric information, if any, related to the address.
Number 2	Displays the numeric information, if any, related to the address.
House Type	Indicates the type of the house.
County	Displays the county name.
State	Indicates the state where the city or county is located.
Postal	Displays the postal or zip code of the address.
In City Limit	Indicates whether the address is within the city limit. The valid values are: • Yes • No
Geographic Code	Displays the geographic code of the address.
Status	Indicates the status of the address. The valid values are: • Active • Inactive

- **Characteristics** - Lists the characteristics defined for the address. It contains the following columns:

Column Name	Column Description
Effective Date	Displays the date from when the characteristic is effective for the address.
Characteristic Type	Indicates the characteristic type.
Characteristic Value	Displays the value of the characteristic type.

- **Address Entities** - Lists the entities associated to the address. It contains the following columns:

Column Name	Column Description
Effective Date	Displays the date from when the address is associated with the entity.
Address Type	Indicates the type of the address. The valid values are: • Account Override • Person Mailing • Person Seasonal
Entity Type	Indicates the type of the entity. The valid values are: • Account • Adjustment • Person
Entity ID	Indicates the entity to which the address is associated. **Note:** It has a link. On clicking the link, the **Person**, **Account**, or **Adjustment** screen appears where you can view the details of the respective entity.
Account-Person ID	Indicates the account holder to whom the address is associated.

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
	Note: The data appears in this column when the entity type is set to **Account** and the address type is set to **Account Override**.

- **Record Actions** - This section contains the following buttons:

Button Name	Button Description
Edit	Used to edit the details of the address.
	Note: If the **Enable Effective Dated Address Feature** option type of the **C1-Address** feature configuration is set to **Y**, you can edit the details of an address which is associated to an entity using the **Person Mailing**, **Person Seasonal**, or **Account Override** address type.
Delete	Used to delete the address.
	Note: You can delete an address when it is not used in either bill routing information of an account or in a statement construct of a person.
Duplicate	Used to create a new address using an existing address.

- **Record Information** - This section contains the following fields:

Field Name	Field Description
Business Object	Indicates the business object using which the address is created. In addition, a context menu appears corresponding to this field which helps in navigating to other screens in the application.
	Note: It has a link. On clicking the link, the **Business Object** screen appears where you can view the details of the respective business object.
Create Date/Time	Displays the date and time when the address is created.

Related Topics

For more information on...	See...
How to edit an address	Editing an Address on page 125
How to delete an address	Deleting an Address on page 129
How to copy an address	Copying an Address on page 127

Editing an Address

Prerequisites

To edit an address, you should have:

- Countries and their states defined in the application
- House types defined in the **HOUSE_TYPE** lookup field

Procedure

To edit an address:

1. Search for the address in the **Address** screen.

Copyright © 2024, Oracle and/or its affiliates.

2. In the **Search Results** section, click the link in the **Address Information** column corresponding to the address whose details you want to edit.
   The **Address Information** screen appears.
3. Ensure that the **Main** tab is selected.
4. Click the **Edit** button in the **Address** zone.

   The **Address** screen appears. It contains the following sections:

   - **Main** - Used to specify basic details about the address.
   - **Characteristics** - Used to define the characteristics for the address.
   - **Address Entities** - Used to associate the address with one or more entities, such as person, account, or adjustment.

   The **Main** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Country	Used to indicate the country. **Note:** The list includes only those countries which are already defined in the system. On selecting a country, only those address fields which are set as **Optional** while defining the country appear in the **Main** section. Also, the labels of the address fields will change depending on the business label which is defined for the respective address field in the **Country** screen.	Yes
State	Used to indicate the state where the city or county is located. **Note:** The states listed in this field change depending on the country that you have selected.	No
Address 1	Used to specify the first line of the address. It may contain details, such as house number and apartment name.	No
Address 2	Used to specify the second line of the address. It may contain details, such as street name.	No
Address 3	Used to specify the third line of the address. It may contain the landmark details.	No
Address 4	Used to specify the fourth line of the address.	No
City	Used to specify the city name.	No
Number 1	Used to specify the numeric information, if any, related to the address.	No
Number 2	Used to specify the numeric information, if any, related to the address.	No
County	Used to specify the county name.	No
Postal	Used to specify the postal or zip code of the address.	No
House Type	Used to indicate the type of the house.	No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: The list includes only those values which are defined in the **HOUSE_TYPE** lookup field.	
In City Limit	Used to indicate whether the address is within the city limit. The valid values are: • Yes • No	No
Geographic Code	Used to specify the geographic code of the address.	No

5. Modify the required details in the **Main** section.

> **Note:** If the **Enable Effective Dated Address Feature** option type of the **C1-Address** feature configuration is set to **Y**, you can edit the details of an address which is associated to an entity using the **Person Mailing**, **Person Seasonal**, or **Account Override** address type.

6. Define, edit, or remove characteristics of the address, if required.
7. Associate or dissociate entities from the address, if required.

> **Note:** If the **Enable Effective Dated Address Feature** option type of the **C1-Address** feature configuration is set to **Y**, you can edit the entity details or disassociate an entity from an address when the address is associated to the entity using the **Person Mailing**, **Person Seasonal**, or **Account Override** address type.

8. Click **Save**.
   The changes made to the address are saved.

> **Note:**
>
> If the approval workflow is configured for the **C1-Address** business object, the approval transaction is created on clicking the **Save** button. Once the approver approves the approval transaction, the changes made to the address are saved.
>
> You can edit an address when it is not used in either bill routing information of an account or in a statement construct of a person.

Related Topics

For more information on...	See...
How to search for an address	Searching for an Address on page 116
Address Information screen	Address Information on page 123
Address zone	Address on page 123
How to define a characteristic for an address	Defining a Characteristic for an Address on page 120
How to associate an address to an entity	Associating an Address to an Entity on page 121

Copying an Address

Instead of creating an address from scratch, you can create a new address using an existing address. This is possible through copying an address. On copying an address, the details including the characteristics and entities are copied to the new address. You can then edit the details, if required.

Copyright © 2024, Oracle and/or its affiliates.

Prerequisites

To copy an address, you should have:

- Address (whose copy you want to create) defined in the application
- Countries and their states defined in the application
- House types defined in the **HOUSE_TYPE** lookup field

Procedure

To copy an address:

1. Search for the address in the **Address** screen.
2. In the **Search Results** section, click the link in the **Address Information** column corresponding to the address whose copy you want to create.
   The **Address Information** screen appears.
3. Click the **Duplicate** button in the **Address** zone.

   The **Address** screen appears. It contains the following sections:

   - **Main** - Used to specify basic details about the address.
   - **Characteristics** - Used to define the characteristics for the address.
   - **Address Entities** - Used to associate the address with one or more entities, such as person, account, or adjustment.

   The **Main** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Country	Used to indicate the country. **Note:** The list includes only those countries which are already defined in the system. On selecting a country, only those address fields which are set as **Optional** while defining the country appear in the **Main** section. Also, the labels of the address fields will change depending on the business label which is defined for the respective address field in the **Country** screen.	Yes
State	Used to indicate the state where the city or county is located. **Note:** The states listed in this field change depending on the country that you have selected.	No
Address 1	Used to specify the first line of the address. It may contain details, such as house number and apartment name.	No
Address 2	Used to specify the second line of the address. It may contain details, such as street name.	No
Address 3	Used to specify the third line of the address. It may contain the landmark details.	No
Address 4	Used to specify the fourth line of the address.	No
City	Used to specify the city name.	No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
Number 1	Used to specify the numeric information, if any, related to the address.	No
Number 2	Used to specify the numeric information, if any, related to the address.	No
County	Used to specify the county name.	No
Postal	Used to specify the postal or zip code of the address.	No
House Type	Used to indicate the type of the house. **Note:** The list includes only those values which are defined in the **HOUSE_TYPE** lookup field.	No
In City Limit	Used to indicate whether the address is within the city limit. The valid values are: • Yes • No	No
Geographic Code	Used to specify the geographic code of the address.	No

4. Enter the required details in the **Main** section.
5. Define, edit, or remove characteristics of the address, if required.
6. Associate or dissociate entities from the address, if required.
7. Click **Save**.
   The address is defined and the address ID is generated.

> **Note:** If the approval workflow is configured for the **C1-Address** business object, the approval transaction is created on clicking the **Save** button. Once the approver approves the approval transaction, the address is defined and the address ID is generated.

Related Topics

For more information on...	See...
How to search for an address	Searching for an Address on page 116
Address Information screen	Address Information on page 123
Address zone	Address on page 123
How to define a characteristic for an address	Defining a Characteristic for an Address on page 120
How to associate an address to an entity	Associating an Address to an Entity on page 121

Deleting an Address

Procedure

To delete an address:

1. Search for the address in the **Address** screen.
2. In the **Search Results** section, click the link in the **Address Information** column corresponding to the address that you want to delete.
   The **Address Information** screen appears.

Copyright © 2024, Oracle and/or its affiliates.

3. Click the **Delete** button in the **Address** zone.
   A message appears confirming whether you want to delete the address.

> **Note:**
>
> You can delete an address when it is not used in either bill routing information of an account or in a statement construct of a person.
>
> You cannot delete an address which is associated to an entity using the **Person Mailing**, **Person Seasonal**, or **Account Override** address type.

4. Click **OK**.
   The address is deleted.

Related Topics

For more information on...	See...
How to search for an address	Searching for an Address on page 116
Address Information screen	Address Information on page 123
Address zone	Address on page 123

Address Information - Log

The **Log** tab contains the following zone:

- Address Log on page 130

Address Log

The **Address Log** zone lists the complete trail of actions performed on the address. It contains the following columns:

Column Name	Column Description
Date/Time	Displays the date and time when the action was performed on the address.
Details	Displays the details about the action performed on the address.
User	Indicates the user who has performed the action on the address.
Log Type	Indicates the type of log.
Related Object	Indicates the object or entity which is created when the action is performed on the address.

You can manually add a log entry for the address by clicking the **Add Log Entry** link in the upper right corner of the **Address Log** zone.

Related Topics

For more information on...	See...
How to view the log of an address	Viewing the Log of an Address on page 130
How to add a log entry for an address	Adding a Log Entry for an Address on page 131

Viewing the Log of an Address

Procedure

Copyright © 2024, Oracle and/or its affiliates.

To view the log of an address:

1. Search for the address in the **Address** screen.
2. In the **Search Results** section, click the link in the **Address Information** column corresponding to the address whose log you want to view.
   The **Address Information** screen appears.
3. Click the **Log** tab.
   The **Log** tab appears.
4. View the complete trail of actions performed on the address in the **Address Log** zone.

Related Topics

For more information on...	See...
How to search for an address	Searching for an Address on page 116
Address Information screen	Address Information on page 123
Address Log zone	Address Log on page 130

Adding a Log Entry for an Address

Procedure

To add a log entry for an address:

1. Search for the address in the **Address** screen.
2. In the **Search Results** section, click the link in the **Address Information** column corresponding to the address whose log you want to edit.
   The **Address Information** screen appears.
3. Click the **Log** tab.
   The **Log** tab appears.
4. Click the **Add Log Entry** link in the upper right corner of the **Address Log** zone.

   The **Add Address Log** window appears. It contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Address Information	Displays information about the address.	Not applicable
Log Details	Used to specify additional comments for the address.	Yes

5. Enter the comments in the **Log Details** field.
6. Click **Save**.
   The log entry is added in the **Address Log** zone.

Related Topics

For more information on...	See...
How to search for an address	Searching for an Address on page 116
Address Information screen	Address Information on page 123
Address Log zone	Address Log on page 130

Copyright © 2024, Oracle and/or its affiliates.

Start / Stop

The topics in this section describe how to start and stop service.

> **CAUTION:** Order or Start / Stop? Please see Order versus Start / Stop for a discussion of the pros and cons of using the order transaction versus the start / stop transaction to start service.

The Big Picture Of Starting Service

Almost all requests to start service involve a conversation between a customer and a customer service representative (CSR). During this conversation, the following is defined:

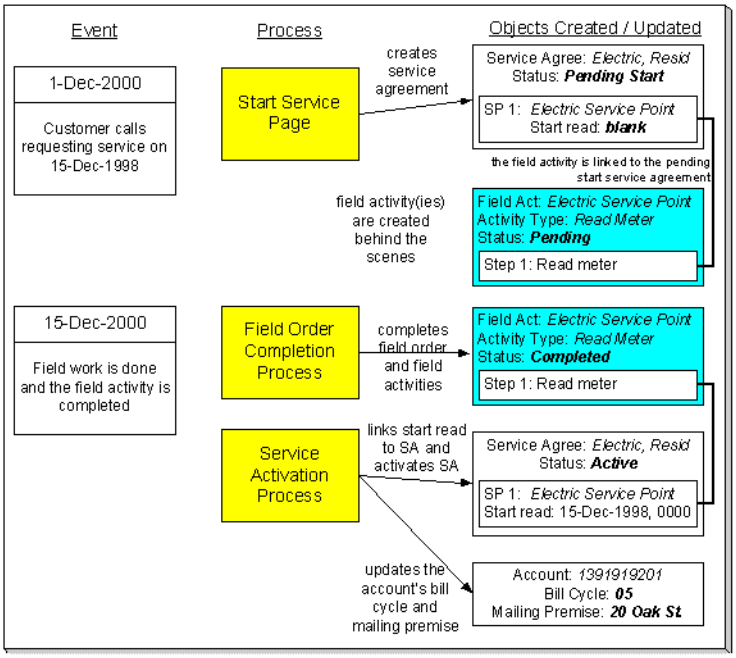
- WHO will pay for the service (i.e., the account).
- WHEN the service is requested (i.e., the start date).
- WHAT TYPE of service is desired (i.e., the contract).
- HOW MUCH will be paid for the service (i.e., the rate and contract riders).

After the conversation is complete, one or more contracts in the state of `pending start` will exist.

In this section, we provide background information about the start process. For information about the page used to start service, refer to Start Stop Maintenance.

The Lifecycle Of A Pending Start Contract

When you request a new service for a customer, a pending start contract is created. The following diagram illustrates the lifecycle of a typical pending start contract:



The above diagram is referenced in the topics that follow.

Activating Pending Starts

When you initiate a request to start service, the system initiates a contract in the state of `pending start`. A pending start contract remains in this state until everything necessary to start service is defined in the system. At that time, the contract can be activated.

Copyright © 2024, Oracle and/or its affiliates.

What Is Necessary To Start Service?

The contract's contract type defines what is necessary to start service. Everything necessary to activate the contract is defined when you initiate the request to start service.

The System Activates Most Contracts Behind-The-Scenes

Contracts are activated by one of the following methods:

- The contract activation background process (referred to by the batch control ID of SAACT) automatically activates `pending start` contracts once it detects that everything necessary to start service is defined in the system. This is the most common method for activating contracts. This process runs as often as your organization desires (but it should run at least daily).
- You can automatically activate a contract when it's created if you plug-in an contract creation algorithm of type SACR-AT on the contract type. Because everything necessary to start service must be defined in the system before a contract can be activated. If this algorithm is not plugged in on an contract type, contracts of that type are activated by the contract activation background process or manually.
- You can activate a contract manually (assuming everything that's necessary to activate service is in place). Simply open Contract - Main and click the **Activate Contract** button. This is only necessary if you do not want to wait until the contract activation background process next runs.

What Happens At Activation Time?

When a contract is activated, the following takes place:

- The contract's state is changed from `pending start` to `active`.
- If the account's division is not protected, the division is changed to reflect the contract's contract type's division.
- The main customer's billing address is changed to reflect the value specified on the Start/Stop page (and this may involve changing the main customer's override mailing address on their person record).

What Happens After A Contract Is Started?

Fastpath: Refer to The Lifecycle Of A Contract for more information about a contract's lifecycle.

The Big Picture Of Stopping Service

To stop service you must define:

- WHO pays for the service (i.e., the account).
- WHEN service should stop (i.e., the stop start date).
- WHAT services to stop (i.e., the contracts).

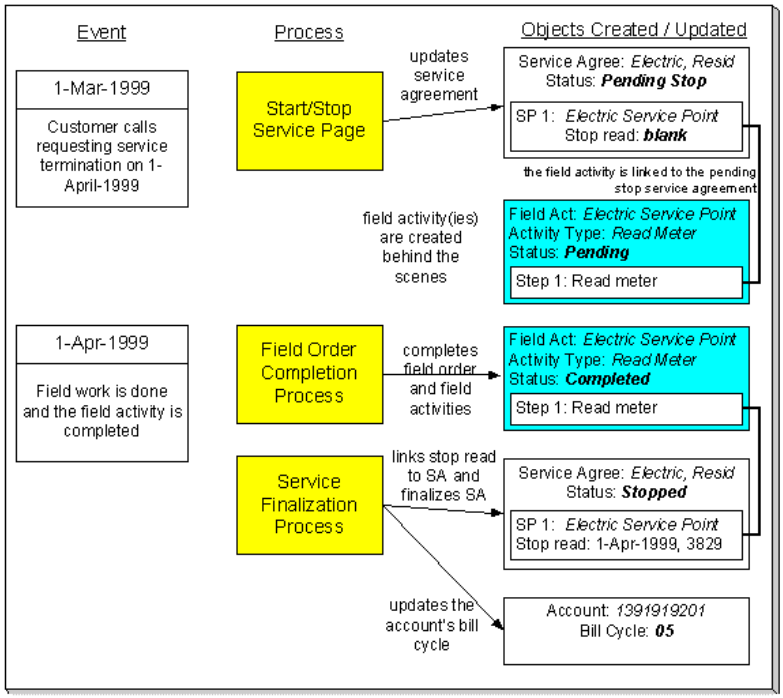
After the above is defined in the system, the state of one or more contracts will be changed to `pending stop`.

In this section, we provide background information about the stop process. For information about the page used to stop service, refer to Start Stop Maintenance.

The Lifecycle Of A Pending Stop Contract

When you request to stop service for a customer, the status of an existing contract becomes `pending stop`. The following diagram illustrates the lifecycle of a typical pending stop contract:

Copyright © 2024, Oracle and/or its affiliates.



The above diagram is referenced in the topics that follow.

Finalizing Pending Stops

When you initiate a request to stop service, the system changes the contract's state to `pending stop`. A `pending stop` contract remains in this state until everything necessary to stop service is defined in the system. At that time, the system automatically changes the state of the contract to `stopped` , and at that time the contract can be final billed.

So, how does the system know that "everything necessary" is in the system? In this section we provide the answer and explain the many other things that may take place at `pending stop` finalization.

The System Finalizes Most Contracts Behind-The-Scenes

The contract activation background process (referred to by the batch control code SAACT) finalizes a `pending stop` contract when everything necessary is in place. Everything necessary to finalize the contract is defined when you initiate the request to stop service.

When the contract activation background process detects that everything necessary to stop service is in place, it automatically `stops` the contract. This process runs as often as your organization desires (but it should run at least daily).

You can stop a contract real-time (assuming everything that's necessary to stop service is in place). Simply open Contract - Main and click the **Stop** button.

What Happens At Finalization Time?

When a contract is finalized, the following takes place:

- If the contract's contract type has a Contract Stop algorithm, the algorithm is executed.
- The contract's state will be changed from `pending stop` to `stopped`.
- If the contract is linked to an `active` collection process, the contract is "removed" from the collection process. If the collection process has no other active contracts, it is `canceled`.
- When the last contract linked to an account is `stopped`, the account's bill cycle will be changed so that the account will be final billed when billing next executes. The system selects the latest open bill cycle where the Freeze and Complete switch value matches the value of the Freeze and Complete switch of the account's current bill cycle.
- The main customer's billing address will be changed to reflect the value specified on the Start/Stop page (and this may involve changing the main customer's override mailing address on their person record).

Copyright © 2024, Oracle and/or its affiliates.

- If the contract is for a deposit (i.e., the contract type has a special role of `Cash Deposit`), the system calculates interest through the stop date. In addition, the contract's current balance is synchronized with its payoff balance (using the sync adjustment code defined on the contract's contract type).
- When the last active contract governed by a deposit class is `stopped`, all `active` deposit contracts governed by the same deposit class will be stopped (i.e., their state will be changed from `active` to `pending stop`). These `pending stop` deposit contract will then be stopped (as per the above points).
- When a contract is stopped, the system automatically cancels the budget.

What Happens After A Contract Is Stopped?

Fastpath: Refer to The Lifecycle Of A Pending Stop Contract on page 133for more information about a contract's lifecycle.

Profile

You can use a profile to create an invoice account while defining a construct. In a profile, you can define the default values for fields related to an entity. This helps you to quickly create an entity using the profile. Note that, at present, you can only define a profile for creating an invoice account.

While defining a construct, the default values appear (depending on how the UI map is defined for creating an entity) when you create an invoice account using the profile. You can edit these values, if required. If the fields are displayed on the UI map, you can edit the values through the UI map or through the respective base entity screen (once the entity is created). However, if the fields are not displayed on the UI map, you can edit the values only through the respective base entity screen (once the entity is created).

You can also associate algorithms with a profile. While associating an algorithm with a profile, you must indicate the system event on which you want to execute the algorithm. You can associate algorithms on the following system events:

- **Account Creation** - Used to attach an algorithm which is triggered when you create an invoice account using the profile.
- **Contract Creation** - Used to attach an algorithm which is triggered when contracts are created for an invoice account which is created using the profile.

Note: The invoice account and contract(s) are created for a construct when the construct is approved by the approver.

You can only use profiles which are active while defining or editing a construct. The **Profile** screen allows you to define, edit, and delete a profile. You can edit a profile even if it is used in any construct. However, any changes made to a profile will not be reflected in a construct where the profile is already used unless you create the account using the latest profile. This screen consists of the following zones:

- Search Profile on page 135

Search Profile

The **Search Profile** zone allows you to search for a profile using various search criteria. It contains the following two sections:

- **Search Criteria** - The **Search Criteria** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Profile	Used to specify the profile name.	No
Description	Used to specify the description for the profile.	No
Entity	Used to indicate the entity for which the profile is created. The valid values are:	No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	• Account	
Status	Used to indicate the status of the profile. The valid values are: • Active • Inactive	No

Note: You must specify at least one search criterion while searching for a profile.

- **Search Results** - On clicking the **Search** button, the search results appear based on the specified search criteria. The **Search Results** section contains the following columns:

Column Name	Column Description
Profile	Displays the profile name.
Description	Displays the description of the profile.
Entity	Indicates the entity for which the profile is created. The valid values are: • Account
Status	Indicate the status of the profile. The valid values are: • Active • Inactive
View	On clicking the **View** (🔗) icon in the **View** column, the **Profile** screen appears where you can view the details of the profile.
Edit	On clicking the **Edit** (✎) icon in the **Edit** column, the **Profile** screen appears where you can edit the details of the profile.
	Note: If a profile is used in any construct, a message appears confirming whether you want to edit the profile.
Delete	On clicking the **Delete** (🗑) icon in the **Delete** column, you can delete the profile.
	Note: You can only delete a profile which is not yet used in any construct.

You can create a new profile by clicking the **Add** button in the upper right corner of this zone.

Related Topics

For more information on...	See...
How to search for a profile	Searching for a Profile on page 136
How to define a profile	Defining a Profile on page 137
How to edit a profile	Editing a Profile on page 140
How to delete a profile	Deleting a Profile on page 142
How to view the details of a profile	Viewing the Profile Details on page 142

Searching for a Profile

Procedure

Copyright © 2024, Oracle and/or its affiliates.

To search for a profile:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **P** and then click **Profile**.
   The **Profile** screen appears.
3. Enter the search criteria in the **Search Profile** zone.

   Note: ORMB search engine supports wildcard search, where you can substitute the percentage (%) symbol as a stand in for any word or letter in a search criteria. You can use the '%' wildcard character in all input fields except the date and ID fields. The '%' wildcard character is suffixed automatically at the end of the partial search criteria. Therefore, you may or may not specify the wildcard character at the end of the partial search criteria. However, you have to prefix the wildcard character manually wherever required.
4. Click **Search**.
   A list of profiles that meet the search criteria appears in the **Search Results** section.

Related Topics

For more information on...	See...
Profile screen	Profile on page 135
Search Profile zone	Search Profile on page 135

Defining a Profile

Procedure

To define a profile:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin Menu**, select **P** and then click **Profile**.
   A sub-menu appears.
3. Click the **Add** option from the **Profile** sub-menu.

   The **Profile** screen appears. It consists of the following tabs:

   - **Main** - Used to specify basic details about the profile.
   - **Algorithms** - Used to indicate algorithms that must be triggered on various system events when you use the profile for creating an entity.

   The **Main** tab contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Profile	Used to specify the profile name.	Yes
Description	Used to specify the description for the profile.	Yes
Entity	Used to indicate the entity for which you want to create a profile. The valid values are: • Account	Yes
Status	Used to indicate the status of the profile. The valid values are: • Active • Inactive	Yes

Copyright © 2024, Oracle and/or its affiliates.

In addition, this tab contains two sections:

- **Default Values** : In this section you can define the default values for fields related to the entity. These default values appear (depending on how the UI map is defined for creating an entity) when you create the entity (for example, an invoice account) using the profile. You can edit these values, if required. If the fields are displayed on the UI map, you can edit the values through the UI map or through the respective base entity screen (once the entity is created). However, if the fields are not displayed on the UI map, you can edit the values only through the respective base entity screen (once the entity is created). The **Default Values** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Field Name	Used to indicate the field for which you want to define the default value.	Yes (Conditional) **Note:** This field is required when you are defining the default value for a field.
Value	Used to specify the default value for the field.	Yes (Conditional) **Note:** This field is required when you are defining the default value for a field.

- **Characteristics** : In this section you can define characteristics of the profile.

Tip: Alternatively, you can access the **Profile** screen by clicking the **Add** button on the **Profile** screen.

4. Enter the required details in the **Main** tab.
5. If you want to define the default value for more than one field, click the **Add** (+) icon in the **Default Values** section and then enter the required details.

   Note: However, if you want to remove the default value of a field from the **Default Values** section, click the **Delete** (🗑) icon corresponding to the field.
6. Associate algorithms with the profile, if required.
7. Click **Save**.
   The profile is defined.

Related Topics

For more information on...	See...
Profile screen	Profile on page 135
Search Profile zone	Search Profile on page 135
How to associate algorithms with a profile	Associating Algorithms with a Profile on page 139
How to define a characteristic for a profile	Defining a Characteristic for a Profile on page 138

Defining a Characteristic for a Profile

Prerequisites

To define a characteristic for a profile, you should have:

- Characteristic types defined in the application (where the characteristic entity is set to Account)

Procedure

To define a characteristic for a profile:

Copyright © 2024, Oracle and/or its affiliates.

1. Ensure that the **Characteristics** section is expanded when you are defining or editing a profile.

   the **Characteristics** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Effective Date	Used to specify the date from when the characteristic is effective for the profile.	Yes (Conditional) **Note:** This field is required when you are defining a characteristic for the profile.
Characteristic Type	Used to indicate the characteristic type. **Note:** The list includes only those characteristic types where the characteristic entity is set to Account.	Yes (Conditional) **Note:** This field is required when you are defining a characteristic for the profile.
Characteristic Value	Used to specify the value for the characteristic type. **Note:** On specifying the value for a predefined characteristic type, the description of the characteristic value appears corresponding to the **Characteristic Value** field.	Yes (Conditional) **Note:** This field is required when you are defining a characteristic for the profile.

2. Enter the required details.
3. If you want to define more than one characteristic for the profile, click the **Add** (+) icon and then repeat step *2*.

> **Note:** However, if you want to remove a characteristic from the profile, click the **Delete** (🗑) icon corresponding to the characteristic.

Related Topics

For more information on...	See...
How to define a profile	Defining a Profile on page 137
How to edit a profile	Editing a Profile on page 140

Associating Algorithms with a Profile

Prerequisites

To associate algorithms with a profile, you should have:

- Algorithms defined for the Account Creation system event using the C1-ACTCR-IC algorithm type
- Algorithms defined for the Contract Creation system event using the C1-CTMP-IC algorithm type

Procedure

To associate algorithms with a profile:

1. Click the **Algorithms** tab in the **Profile** screen when you are creating or editing a profile.

   The **Algorithms** tab appears. It contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Sequence	Used to specify the order in which the algorithms for each system event must be executed.	Yes (Conditional)

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
		Note: This field is required when you are associating an algorithm with the profile.
System Event	Used to indicate the system event on which the algorithm must be executed. The valid values are: • **Account Creation** - Used to attach an algorithm which is triggered when you create an invoice account using the profile. • **Contract Creation** - Used to attach an algorithm which is triggered when contracts are created for an invoice account which is created using the profile.	Yes (Conditional) **Note:** This field is required when you are associating an algorithm with the profile.
Algorithm	Used to specify the algorithm code.	Yes (Conditional)
	Note: On specifying the algorithm code, the description of the algorithm appears corresponding to the **Algorithm** field. The **Search** (🔍) icon appears corresponding to this field. On clicking the **Search** icon, the**Algorithm Search** window appears.	**Note:** This field is required when you are associating an algorithm with the profile.

2. Enter the required details.
3. If you want to associate more than one algorithm with the profile, click the **Add** (+) icon and then repeat step *2*.

Note: However, if you want to remove an algorithm from the profile, click the **Delete** (🗑) icon corresponding to the algorithm.

4. Click **Save**.
   The algorithms are associated with the profile.

Related Topics

For more information on...	See...
How to define a profile	Defining a Profile on page 137
How to edit a profile	Editing a Profile on page 140

Editing a Profile

Procedure

To edit a profile:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **P** and then click **Profile**.
   The **Profile** screen appears.
3. In the **Search Results** section, click the **Edit** (✎) icon in the **Edit** column corresponding to the profile whose details you want to edit.

Copyright © 2024, Oracle and/or its affiliates.

> **Note:** If a profile is used in any construct, a message appears confirming whether you want to edit the profile. You must click **OK** if you want to edit the profile. Any changes made to a profile will not be reflected in any construct where the profile is already used unless you create the entity using the latest profile.

The **Profile** screen appears. It consists of the following tabs:

- **Main** - Used to specify basic details about the profile.
- **Algorithms** - Used to indicate algorithms that must be triggered on various system events when you use the profile for creating an entity.

The **Main** tab contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Profile	Displays the profile name.	Not applicable
Description	Used to specify the description for the profile.	Yes
Entity	Used to indicate the entity for which you want to create a profile. The valid values are: • Account	Yes
Status	Used to indicate the status of the profile. The valid values are: • Active • Inactive	Yes

In addition, this tab contains two sections:

- **Default Values** : In this section you can define the default values for fields related to the entity. These default values appear (depending on how the UI map is defined for creating an entity) when you create the entity (for example, an invoice account) using the profile. You can edit these values, if required. If the fields are displayed on the UI map, you can edit the values through the UI map or through the respective base entity screen (once the entity is created). However, if the fields are not displayed on the UI map, you can edit the values only through the respective base entity screen (once the entity is created). The **Default Values** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Field Name	Used to indicate the field for which you want to define the default value.	Yes (Conditional) **Note:** This field is required when you are defining the default value for a field.
Value	Used to specify the default value for the field.	Yes (Conditional) **Note:** This field is required when you are defining the default value for a field.

- **Characteristics** : Define, edit or remove characteristics from the template, if required.

4. Modify the details in the **Main** tab, if required.
5. If you want to define the default value for more than one field, click the **Add** (+) icon in the **Default Values** section and then enter the required details.

> **Note:** However, if you want to remove the default value of a field from the **Default Values** section, click the **Delete** (🗑) icon corresponding to the field.

Copyright © 2024, Oracle and/or its affiliates.

6. Associate algorithms with the profile, if required.
7. Click **Save**.
   The changes made to the profile are saved.

Related Topics

For more information on...	See...
How to search for a profile	Searching for a Profile on page 136
How to define a characteristic for a profile	Defining a Characteristic for a Profile on page 138
How to associate algorithms with a profile	Associating Algorithms with a Profile on page 139

Deleting a Profile

Procedure

To delete a profile:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **P** and then click **Profile**.
   The **Profile** screen appears.
3. In the **Search Results** section, click the **Delete** (🗑) icon in the **Delete** column corresponding to the profile that you want to delete.

   A message appears confirming whether you want to delete the profile.

   Note: You can only delete a profile which is not yet used in any construct.
4. Click **OK**.
   The profile is deleted.

Related Topics

For more information on...	See...
How to search for a profile	Searching for a Profile on page 136

Viewing the Profile Details

Procedure

To view the details of a profile:

1. Search for the profile in the **Profile** screen.
2. In the **Search Results** section, click the **View** (🔗) icon in the **View** column corresponding to the profile whose details you want to view.
   The **Profile** screen appears.
3. View the basic details of the profile in the **Main** tab.
4. View the algorithms associated with the profile in the **Algorithms** tab.

Related Topics

For more information on...	See...
How to search for a profile	Searching for a Profile on page 136

Copyright © 2024, Oracle and/or its affiliates.

Start/Stop Maintenance

You use the start/stop page to both start and stop service.

This page handles a variety of scenarios, including:

- Starting services.
- Stopping services.
- Changing the mailing address associated with the account's main customer.

Fastpath: Refer to The Big Picture Of Starting Service for more information about starting service. Refer to The Big Picture Of Stopping Service for more information about stopping service.

The topics in the section describe the start/stop page in detail.

Start/Stop - Main

Fastpath: The **Description of Page** section that appears below simply describes the fields on this page. Refer to How To for a description of how to perform common start/stop functions.

Use **Person Information, Start/Stop** to open this page.

Description of Page

The **Account ID** and the primary name of the **Account**'s main customer are displayed at the top of the page.

If the total number of pending start / stops for the account does not exceed the start stop details threshold defined in Installation Options, the grid contains an entry for every `Pending Start` and `Pending Stop` contract linked to the **Account ID**.

The following columns appear in the grid:

Drill Down (button) Click this button to drill down into details of the pending start / stop contract. This button transfers you to the **Pending Contract** tab.

Task An icon will be displayed to highlight if the associated contract is a pending start or a pending stop. The icon appears if the contract is a pending start. The icon appears if the contract is a pending stop.

Date This is the start date for pending start contracts and the stop date for pending stop contracts.

Cancel Turn on this switch if you wish to cancel the pending start / stop.

Location Information If a characteristic location is linked to the contract, summary information about the pending start / stop contract's characteristic location is displayed.

Contract Information Summary information about the pending start / stop contract is displayed.

If the total number of pending start / stops for the account exceeds the start stop details threshold defined in Installation Options, the **Total** number of **Pending Starts** and **Total** number of **Pending Stops** is displayed together with a grid containing a summary of the pending start and pending stops.

The grid contains an entry for every group of `Pending Start`/ `Pending Stop` contract linked to the **Account ID** with the same pending start / pending stop date, action, location, and division/contract type.

The following columns appear in the grid:

Date This is the start date for pending start contracts and the stop date for pending stop contracts.

Task An icon will be displayed to highlight if the associated contract(s) are pending start or pending stop. The icon appears if the contract(s) are a pending start. The icon appears if the contract(s) are a pending stop. Click on the icon to navigate to the Start Stop - Pending Contract tab where the associated pending start / pending stop contracts will be displayed.

Division / Contract Type This is the division and contract type of the pending start / pending stop contracts.

Copyright © 2024, Oracle and/or its affiliates.

Total This is the total number of contracts with the same **Date**, **Task**, **Location**, and **Division / Contract Type**.

Following the `Pending Start / Pending Stop` grid are two sections; one is used to start service, the other is used to stop service.

The following points describe the fields in the *start* service section (refer to How To for a description of how to perform common start functions).

- Use **Start Date** to define the date on which you want service to commence.
- Use **Start Method** to define what you want to start. Valid values are:
  - `Start a Contract`. Use this method if you want to start deposit or one time charge service.
- If you have selected the `Start a Contract` option, use **Division** and **Contract Type** to define the type of contract you wish to start.

CAUTION: Take special care when adding a new contract to specify the appropriate division / contract type as it will affect how the customer is billed, what rate can be specified, how overdue debt is collected, and much more. After the contract is activated, you may not change its contract type.

- Use **Requested by** to define who requested to start the service (e.g., `Customer`, `Jane Smith`).
- Click the **Start** button to start service. When you click this button, the Start Confirmation page appears.

The following points describe the fields in the *stop* service section (refer to How To for a description of how to perform common stop functions).

- Use **Stop Date** to define the date on which you want service to stop.
- Use **Stop Method** to define what you want to stop. Valid values are:
  - `Stop Selected Contracts`. Use this method if you want to stop selected contracts linked to the account.
  - `Stop a Contract`. Use this method if you want to stop a specific contract.
  - `Stop all Active Contracts`. Use this method if you want to stop ALL contracts linked to the account.
- If you have selected the `Stop a Contract` option, choose the **Contract ID**.
- Use **Requested by** to define who requested the stop (e.g., `Customer`, `Jane Smith`).
- Click the **Stop** button to stop service. When you click this button, the Stop Confirmation page appears.

Start/Stop - Start Confirmation

The Start Contract Confirmation page appears after a user clicks the **Start** button on Start/Stop - Main. This page presents a summary of the contract(s) that will be created if the **Start** button on this page is clicked. Refer to How To for a description of how to perform common start functions.

Description of Page

The **Account ID** and the primary name of the account's main customer are displayed at the top of the page.

The **Start Date** is the date on which the service is scheduled to start.

Start Requested By contains the name of the person who requested the start.

The **Contracts to Start** scroll contains an entry for every contract that will be created if you click the **Start** button. If you do not want a contract to be created, click the - button (or turn off **Start this Contract**). If you'd like to start a contract, click the + button. The following fields appear on this page:

Start this Contract	Turn this switch on if you want this type of contract to be created. Turn this switch off if you do not want this type of contract to be created. If you used a **Start Method** of **Start a Contract** on the Start/Stop page, this switch will be turned on by default.
Division / Contract Type	These fields define the type of contract that will be created. If you change the division, you must change the

Copyright © 2024, Oracle and/or its affiliates.

	contract type to reflect one of the services that the division manages.
Total Amount to Bill	Use this field if the contract is supposed to bill a given amount over its lifetime. This is typically used for deposit contracts to define the amount of the deposit that should be billed. If the deposit is billed in installments, the installment amount is specified in the next field. This field is gray if the respective contract type doesn't use the Total Amount to Bill option.
	This button is only appears if the contract type has a "special role" of `Cash Deposit` and references a deposit class. When clicked, the system attempts to calculate a recommended deposit amount by calling the recommended deposit amount algorithm that is plugged in on the contract type's deposit class.
Recurring Charge Amount	This field's label is dynamic based on the content of the Recurring Charge Amount Label on the respective contract type. This field is gray if the respective contract type doesn't use the Recurring Charge option. This field is typically used only for the following types of contracts: • Contracts on a levelized payment plan. This field contains the amount charged each month for the service regardless of the actual amount owed. • Deposit contracts on an installment plan. This field contains the installment amount.
Contract Information	This is a display-only area that provides information about the specific pending start contract.
Messages	This is a display-only area that provides information about the account's start / stop situation in general.

When you click the **Start** button, the system performs the following for every service:

- A contract is created effective on the start date.
- The contract's rate is populated with the contract type's default rate.

Start/Stop - Stop Confirmation

If you choose the stop method of `Stop Selected Contracts`, the Stop Contract Confirmation page appears after a user clicks the **Stop** button on Start/Stop - Main. This page presents a summary of the contract(s) that will be stopped if the **Stop** button on this page is clicked. Refer to How To for a description of how to perform common start stop functions.

Description of Page

The **Account ID** and the primary name of the account's main customer are displayed at the top of the page.

The **Stop Date** is the date on which the service is scheduled to stop.

Requested By contains the name of the person who requested the stop.

The section beneath **Requested By** provides you with options that control which contracts appear in the grid. The following points describe the various options:

- Use the **Location Filter** to restrict the contracts to those that are linked to a specific characteristic location. The following options are available:

Copyright © 2024, Oracle and/or its affiliates.

  - `All`. Use this option if you do not wish to restrict contracts based on location.
  - `No Location`. Use this option to restrict contracts to those that do not have a characteristic location.
  - `Specific Location`. Use this option to restrict contracts to those with a characteristic location of **Location ID**.
- Use the **Contract Type Filter** to restrict contracts to those of a given contract type.
  - `All`. Use this option if you do not wish to restrict contracts based on contract type.
  - `Specific Contract Type`. Use this option to restrict contracts to those of a given **Division** and **Contract Type**.

Don't forget to click the search button after changing the filters.

The **Select All**/ **Clear All** buttons are used to select contracts to stop. These buttons are enabled if at least one contract is displayed in the grid.

Note: 30 contracts at a time. Clicking **Select All** selects the first 30 contracts in the grid. If more than 30 contracts exist, you must select and stop them in batches.

The grid that follows contains the contracts to stop that match your filter criteria. The following information appears in the grid:

Select Box Use this check box to select contracts that you want to stop.

Contract Information Summary information about the pending stop contract is displayed.

Click the **Stop** button on the Stop Contract Confirmation page to stop the selected contracts. Click on the **Close** button to return to the main page - any changes made on this page since the last time you clicked **Stop** are discarded.

Start/Stop - Address/Phone

Open **Person Information, Start Stop, Address/Phone** to change basic information about a customer while you are starting or stopping service.

Note: A subset of person and account information. This page features a subset of person and account fields that typically change when a customer moves. You can always use the Person or Account transactions to change this information.

Note: This information is held in limbo. The information on this page will only be updated on the respective person / account when the first contract linked to the account is activated or stopped (i.e., this information is held in limbo until the pending start / pending stop is activated / stopped).

Note: Main customer information. The information on this page is associated with the account's main customer (as defined on Account - Person Information). If you need to change information about any other person linked to the account, you must transfer to Account - Person Information and make the changes there.

Use **Bill Route Type** to indicate how the bill is sent to the customer. If the **Bill Route Type** you select indicates that bills are routed via fax, make sure you specify a fax number in the **Telephone Numbers** below (note, the system knows which of a person's phone numbers is a fax number by the phone type). If the **Bill Route Type** you select indicates that bills are routed via Email, make sure to specify the person's **Email Address** below. If the **Bill Route Type** you selected indicates that bills are routed via the postal service, select the appropriate **Address Source**. This field is protected if bills are routed via email or fax. You have the following options:

- Choose `Mailing Location on Account` if bills should be sent to the account's mailing location. This address is displayed below.
- Choose `Person` if bills should be sent to the *person's mailing address*. This address is displayed below.
- Choose `Account Override` if bills should be sent to the account person's override address. This address is displayed below. Typically, you would only choose this option if the person has multiple accounts and each account's bills should be sent to a different address.

Copyright © 2024, Oracle and/or its affiliates.

If the **Address Source** is `Person` or `Account Override`, the respective address fields are displayed and may be changed. If the **Address Source** is `Mailing Location on Account`, the account's mailing location is displayed and cannot be modified.

Note: If you switch the **Bill Route Type** from a non-postal to postal routed, the **Address Source** defaults to the bill address source defined on the account (if one exists) or `Mailing Location on Account` if bill address source is not specified for the main customer.

Fastpath: Refer to Setting Up Bill Route Types for more information about bill route types.

Specify the customer's **Email Address**. This can be used to route bills to the customer via email (if your bill routing software supports this function).

Telephone Numbers are used to look for accounts and persons. If the person has more than 20 phone numbers, the first 20 will be shown as display only with a message indicating that more phone numbers exist until you save your start or stop. Once the start or stop has been saved, you are able to modify the phone number collection.

The following fields display:

- **Phone Type** indicates the type of phone number, e.g., Home, Mobile, Business,...
- Use **Phone Number** to define the telephone number. Enter the telephone number in the format described by the **Phone Format**.

Note: Formatting is performed by a plug-in. The format that is applied to a **Phone Number** is controlled by the algorithm that is plugged in on the respective Phone Type. If you prefer a different format, your system administrator should configure this algorithm appropriately.

- Enter the **Extension**, if any, of the telephone number.

Start/Stop - Pending Contract

If you need to change additional information about the pending start or stop contracts, navigate to the **Pending Contract** tab. Alternatively, you can click the Drill Down button on the Contract grid of the **Main** tab.

Description of Page

If the total number of pending start / stops for the account does not exceed the start stop details threshold defined in Installation Options, an entry appears in the scroll for each entry in the Contract grid on the main page. Use the **Pending Start/Stop Contract** navigation arrow buttons to select the pending contract you want to review or modify. The following fields are available.

Primary Contract:

Contract Info Summary information about the pending start / stop contract is displayed.

Cancel Turn on this switch if you wish to cancel the pending start / stop.

Date This is the start date for pending start contracts and the stop date for pending stop contracts.

Reason Indicates the reason for starting or stopping a service. .

Division / Contract Type These fields define the type of contract that will be created. These fields will be gray if the contract is pending stop.

If you change the division, you must change the contract type to reflect one of the services that the division manages.

If the total number of pending start / stops for the account exceeds the start stop details threshold defined in Installation Options, the pending start / stop contracts are displayed in a grid; you can control the contracts that appear using filter options.

The section beneath **Account** provides you with options that control which contracts appear in the grid. The following points describe the various options:

Copyright © 2024, Oracle and/or its affiliates.

- Use the **Contract Action Filter** to restrict the contracts that appear in the grid to pending starts or pending stops. The following options are available:
  - `All`. Use this option if you do not wish to restrict contracts based on Contract Action.
  - `Start`. Use this option to restrict contracts to pending starts.
  - `Stop`. Use this option to restrict contracts to pending stops.
- Use the date filter to restrict contracts to those that will start or stop between the **From Date** and the **To Date**.
- Use the **Location Filter** to restrict the contracts to those that are linked to a specific characteristic location. The following options are available:
  - `All`. Use this option if you do not wish to restrict contracts based on location.
  - `No Location`. Use this option to restrict contracts to those that do not have a characteristic location.
  - `Specific Location`. Use this option to restrict contracts to those with a characteristic location of **Location ID**.
- Use the **Contract Type Filter** to restrict contracts to those of a given contract type.
  - `All`. Use this option if you do not wish to restrict contracts based on contract type.
  - `Specific Contract Type`. Use this option to restrict contracts to those of a given **Division** and **Contract Type**.

Don't forget to click the search button after changing the filters.

The **Select All**/ **Clear All** buttons are used to select contracts if you plan on issuing any of the mass update actions at the bottom of the page. These buttons are enabled if at least one contract is displayed in the grid.

Note: 30 contracts at a time. Clicking **Select All** selects the first 30 contracts in the grid. If more than 30 contracts exist, you must select and start them in batches.

The grid that displays contains the contracts that match your search criteria. Click on the drill down button to open the **Pending Contract Details** page.

Please refer to the **Pending Start/Stop Contract** scroll for a description of the fields in the pending start/stop contract grid on the Pending Contract tab and on the Pending Contract Details page. Additional fields are described here:

Pending Contracts Grid:

- **Select box**. Use this check box to select contracts for mass update.
- The **Action** icons highlight if the associated contract is a pending start or a pending stop. Refer to the action icon on Start/Stop - Main.

Mass Update Buttons:

Use the **Cancel Selected** button to cancel the selected pending start / stops. This button is enabled if you have selected one or more pending contracts.

Use the **Change Start/Stop Date** button to change the start / stop date for the selected pending start / stop contracts. This button is enabled if you have selected one or more pending contracts.

How To

The topics in this section describe how to perform common start/stop functions.

How To Start A Deposit

Charging a customer for a deposit involves creating a contract. Deposit contracts are just like other contracts in many ways:

- The system bills the customer for the deposit using information on the contract.
- The customer pays for the billed amount.
- If the customer doesn't pay, credit & collections starts a collection process to encourage the customer to pay.

Deposit contracts differ from regular contracts in the following ways:

Copyright © 2024, Oracle and/or its affiliates.

- Deposit contracts typically don't cause a receivable account to be affected when the deposit is billed. It is when the deposit is paid that a financial transaction occurs (debit cash, credit deposit payable).
- Deposit contracts have a credit balance when the billed amount is paid (the deposit paid by the customer is a liability of your company - it owes the customer this money).
- Deposit contracts may be billed in installments or in a lump sum.

Creating a deposit contract involves the following steps:

- From the **Main Menu**, select **Customer Information** and then click **Start/Stop**. The **Start Stop** screen appears.
- Search the account for which you want to create a deposit contract
- Enter a **Start Date** and **Start Method** of `Start a Contract`.
- Click the **Start** button.
- After you click the **Start** button, the Start Confirmation page will appear:
  - Choose an appropriate **Division / Contract Type** for the deposit contract.
  - Define the total amount of the deposit and the installment amount, if any. You define the total amount of the deposit in **Total Amount To Bill**. You can have the system recommend a deposit amount by pushing the that appears adjacent to this field.
  - If you allow the customer to pay the deposit in installments, define the Installment Amount (the prompt for this field is defined on the contract type and therefore will differ depending on how you've set up the system). After defining this information, close the page. The system will use the information you entered to create a deposit contract.

How To Start A One-Time Charge Contract

You create a one-time charge (i.e., billable charge) whenever a customer should be charged for a service that occurs outside the normal course of business.

A billable charge must reference a contract. This contract behaves just like any other contract:

- **Bill segments are created for the contract.** Whenever billing is performed for an account with billable charge contracts, the system creates a bill segment for each unbilled billable charges.
- **Payments are distributed to the contract.** Payments made by an account are distributed to its billable charge contracts just like any other contract.
- **Overdue debt is monitored.** The credit and collections process monitors billable charge contracts for overdue debt and responds accordingly when overdue debt is detected.

Therefore, in order to levy a one-time charge, the customer must have a billable charge contract. If one already exists (because of a previous billable charge), you don't have to create a new contract. Rather, transfer to the Billable Charge page and add a billable charge for the contract. If the customer doesn't already have billable charge contract, create one by performing the following steps:

- From the **Main Menu**, select **Customer Information** and then click **Start/Stop**. The **Start Stop** screen appears.
- Search the account for which you want to create a one-time charge.
- Enter a **Start Date** and **Start Method** of `Start a Contract`.
- Indicate the **Division**/ **Contract Type** of the billable charge contract and click the **Start** button.
- After you click the **Start** button, the Start Confirmation page will appear. Simply close the confirmation page to save the contract.
- After the billable charge contract has been created, add a billable charge to it (the easiest way is to use the Contract context menu). Refer to Maintaining Billable Charges for more information.

The Big Picture Of Customer Contacts

Customer contacts are used to record when and why a customer contacted your company. This information is used for both audit and statistical purposes. The topics in this section provide details about how customer contacts are created and used in the system.

Copyright © 2024, Oracle and/or its affiliates.

Customer Contacts Are Associated With Persons, NOT Accounts

Customer contacts are associated with a Person, not with an Account. To associate a customer contact record with an account, you must select a person related to that account. Unless you have specific reasons to do otherwise, the main customer is usually your best choice.

How Customer Contacts Are Created

You can add a Customer Contact at any time using either of the following methods:

- Use the customer contact maintenance page.
- Use the Customer Contact Zone on the dashboard.

In addition to manually adding customer contacts, your implementation team can set up plug-in algorithms that create customer contacts when certain events transpire. For example, you can plug-in an algorithm on the customer class object that creates a customer contact whenever an order is completed. Because the number and type of plug-ins can be customized for your implementation, we cannot provide a concise list of all such algorithms.

In addition to the above, a variety of events can trigger the creation of a customer contact. Refer to Customer Contacts Are Used To Trigger Letters for the details.

Customer Contacts Are Used To Trigger Letters

In order to send a letter to a customer, a customer contact must be created for the customer. These types of customer contacts reference a customer contact type that, in turn, references a letter template. The letter template controls the type of information that is merged into the "form letter" and how the letter is physically produced. Refer to Printing Letters for more information about letter templates and how letters are physically produced.

Customer contacts that trigger letters can be produced by any of the method described under How Customer Contacts Are Created. While a user can create this type of customer contact (e.g., if a customer requests a form to sign-up for automatic payment), the primary sources of customer contacts that trigger letters are via system events and algorithms. Besides algorithms, the following system events may generate customer contacts automatically:

- **Collection events.** When a customer falls in arrears, the system creates a collection process. A collection process is a series of events meant to encourage the customer to pay their outstanding debt. A classic collection event is one that creates a customer contact that, in turn, causes a letter to be sent to the customer requesting payment. Refer to The Big Picture Of Collection Events for more information.
- **Write-off events.** After a customer has been final billed, a write-off process will be created. A write-off process is a series of events that leads to the debt being written off. A classic write-off event is one that creates a customer contact that, in turn, causes a letter to be sent to the customer informing them of that their debt has been referred to a collection agency. Refer to The Big Picture Of Write Off Events for more information.

Your implementation team can introduce plug-in algorithms to create a customer contact when certain events take place. If the created customer contact references a letter template, a letter will be triggered. Because the number and type of plug-ins can be customized for your implementation, we cannot provide a concise list of all such algorithms.

A Customer Contact May Trigger Reminders

If you want the system to remind you (via a To Do entry) about a customer-related issue, you can set up a "reminder" on a customer contact. To do this, follow these steps:

- Determine if you need a new customer contact or if you can reuse one that already exists. If you need a new customer contact, use any of the "online methods" described under How Customer Contacts Are Created.
- Add an entry to the customer contact's log. This log entry should be set up as follows:

Copyright © 2024, Oracle and/or its affiliates.

- - If the To Do entry should be addressed to a specific user, choose a **Reminder** type of `Send to User` and enter the user's **User ID**.
  - If the To Do entry should be addressed to a group of users, choose a **Reminder** type of `Send to Role` and enter the user group's **To Do Role**.
  - Use **Trigger Date** to define the *latest* date on which the To Do entry should be created. The reason we indicated this should be the *latest* date is because the background process that's responsible for creating these To Do entries has a parameter called "lead time". This parameter is used to define the number of days before the **Trigger Date** that the To Do entry should be created. Note, the batch control ID of `TD-CCCB` is used to refer to this background process.

A To Do entry will then be created on a future date.

Customer Contacts Can Be Used As Case Files

If a customer calls with a complaint (e.g., they suspect their bill is too high), you can start a "case file" for the complaint by creating a customer contact. This customer contact should be marked as "open".

> **Note: Alerts.** Customer contacts that are marked as being open cause an alert to appear when the customer is displayed on control central.

Over time, as you work on the case, you can add entries to the customer contact's log to describe your progress.

You can use the customer contact's reminder functionality to remind you to check out the case on a future date (or dates).

When the case is resolved, you simply turn off the customer contact's open indicator and add an appropriate entry to the log describing the resolution.

Maintaining Customer Contacts

Customer contacts are used to record when and why a customer contacted your company. This information is used for both audit and statistical purposes. The topics in this section describe how to maintain customer contacts. Refer to The Big Picture Of Customer Contacts for background information about customer contacts.

Customer Contact - Main

Open **Customer Management, Customer Contact** to maintain a customer contact.

> **Note: The dashboard.** A zone exists in the dashboard that can be used to easily create new customer contacts.

Description of Page

Enter the **Person ID** of the person associated with the customer contact.

Select a **Preferred Contact Method** to indicate how the person prefers to be contacted. The following options are available:

- `Email.` If this method is selected, the Contact Person's email address appears adjacent.
- `Fax.` If this method is selected, the Contact Person's fax number appears (the system knows which of a person's phone numbers is a fax number by the phone type).
- `Not Applicable.` If the person doesn't want to be contacted (or the contact method is unusual), use this option. You might want to consider describing the situation in the Contact Instructions.
- `Phone.` If this method is selected, the Contact Person's phone number appears.
- `Postal.` If this method is selected, the Contact Person's address appears. Refer to Where Does The Address Come From for a description of which address the system displays.
- `SMS.` If this method is selected, the Contact Person's short message service (SMS) information displays.

Copyright © 2024, Oracle and/or its affiliates.

Turn on the **Open** switch if the event that necessitated the contact hasn't been resolved. For example, if a customer calls with a high bill complaint and you can't resolve it immediately, you would turn on the **Open** switch and enter an appropriate entry in the **Log**. Refer to Customer Contacts Can Be Used As Case Files for more information.

Enter the **Contact Date** and **Contact Time**.

Note: The current date and time are defaulted.

The **User ID** of the user who created the contact is displayed adjacent to **Contact Date / Time**.

Every customer contact has a **Contact Type** that classifies the record for reporting purposes. Every contact type, in turn, references a **Contact Class**. The class categorizes customer contacts into larger groupings for reporting purposes.

Note: Adding a customer contact may cause a letter to be generated. You can set up a customer contact type to generate a form letter whenever a customer contact of this type is added. In fact, this is the only way to generate a letter in the system. Refer to Customer Contacts Are Used To Trigger Letters for more information.

Use **Comments** to describe the contact.

If the customer contact was created by a collection event or a write-off event, the **Related Records** contains information about the associated record.

If a letter template is associated with the customer contact type / class:

- **Letter Information** describes the status of the letter (i.e., whether it has been printed or not).
- If the letter has been printed and you want to reprint, please turn on the reprint switch (this will cause the letter to be reprinted the next time the respective letter print background process executes).
- If your implementation team has plugged-in an Online Letter Image algorithm on Installation Options - Algorithms, the **Display Letter** button appears. When clicked, the image of the letter is rendered in a PDF and displayed in an Adobe reader. Refer to Technical Implementation Of Online Letter Production for a technical description of how letter images are produced.

The grid that follows contains a diary of past (and future) events related to the customer contact. Refer to A Customer Contact May Trigger Reminders and Customer Contacts Can Be Used As Case Files for more information about when you would use this log.

There are two ways to add a row to the log:

- You can click the + button to add a new row.
- You can navigate to the **Log** tab and insert a new row into the scroll.

Regardless of the method used to add a log entry, the following information appears on a log entry:

- **Create Date / Time** is the date and time when the log entry was created.
- **Created by** identifies the user that created the log entry.
- Use **Comments** to describe the reason for the log entry. If you have lengthy comments, we recommend navigating to the **Log** tab (by clicking the adjacent Go To button) as there is a larger input field on this page.
- The remaining fields are only used if you want the system to remind you about this customer contact on a future date. If you enter these fields, the system will create a To Do entry to remind you about the customer contact.
    - If the To Do entry should be addressed to a specific user, choose a **Reminder** type of `Send to User` and enter the user's **User ID**.
    - If the To Do entry should be addressed to a group of users, choose a **Reminder** type of `Send to Role` and enter the user group's **To Do Role**.
    - Use **Trigger Date** to define the *latest* date on which the To Do entry should be created. The reason we indicated this should be the *latest* date is because the background process that's responsible for creating these To Do entries has a parameter called "lead time". This parameter is used to define the number of days before the **Trigger Date** that the To Do entry should be created. Note, the batch control ID of `TD-CCCB` is used to refer to this background process.

Copyright © 2024, Oracle and/or its affiliates.

> **Note: Multiple reminders.** You can set up multiple reminders on a customer contact. For example, you can indicate that you want to be reminded every Monday for the next 4 weeks to check on the issue that caused a given customer contact to arise. You'd do this be entering 4 log entries where each as the desired **Trigger Date**.

Customer Contact - Log

The log contains a diary of past (and future) events related to the customer contact. Open **Customer Management, Customer Contact** and navigate to the **Log** tab to maintain a customer contact's log. Note, you can also maintain this information on the **Main** tab.

> **Fastpath:** Refer to A Customer Contact May Trigger Reminders and Customer Contacts Can Be Used As Case Files for more information about when you would use this log.

Description of Page

Date / Time is a display-only field that contains the date and time when the log entry was created.

Log ID is the system-assigned, unique identifier of the log entry.

Created by is a display-only field that contains the ID and name of the user who created the log entry.

Use **Comments** to describe the reason for the log entry.

The remaining fields are only used if you want the system to remind you about this customer contact on a future date. If you enter these fields, the system will create a To Do entry to remind you about the customer contact.

- If the To Do entry should be addressed to a specific user, choose a **Reminder** type of `Send to User` and enter the user's ID in **Send To**.
- If the To Do entry should be addressed to a group of users, choose a **Reminder** type of `Send to Role` and enter the user group's To Do Role in **Send To**.
- Use **Trigger Date** to define the *latest* date on which the To Do entry should be created. The reason we indicated this should be the *latest* date is because the background process that's responsible for creating these To Do entries has a parameter called "lead time". This parameter is used to define the number of days before the **Trigger Date** that the To Do entry should be created. Note, the batch control ID of `TD-CCCB` is used to refer to this background process.

Customer Contact - Characteristics

Open **Customer Management, Customer Contact** and navigate to the **Characteristic** tab to maintain a customer contact's characteristics.

> **Note:** We do not anticipate a great deal of use for characteristics on customer contacts. The primary purpose of catering for characteristics on this object is to allow a customer contact that triggers a letter to be able to extract information from any object in the system.

Description of Page

> **Note:** You can only choose characteristic types defined as permissible on a customer contact. Refer to Setting Up Characteristic Types & Their Values for more information about characteristic types. In addition, the characteristic type must also be defined as valid for the customer contact's type. Refer to Setting Up Customer Contact Types for more information about customer contact type.

The following fields display:

Characteristic Type Indicate the type of characteristic.

Characteristic Value Indicate the value of the characteristic.

Copyright © 2024, Oracle and/or its affiliates.

Printing Letters

As described under Customer Contacts Are Used To Trigger Letters, you must create a customer contact whenever you want to send a letter to a customer. The contents of this section describe the technical implementation of letter production.

Letter Templates Control The Information Merged Onto Letters

Customer contacts that trigger letters reference a customer contact type that, in turn, references a letter template. The letter template controls the following:

- It contains an algorithm that is responsible for extracting the information merged onto your letters. Specifically, algorithms of this type create the "flat file records" that are passed to your letter print software. Algorithms of this type are called under the following scenarios:
  - The background process that builds the flat file that's passed to your letter print software calls these algorithms to construct the "flat file records" for each letter.
  - If your letter print software has the ability to construct a real-time image of a letter (in a PDF), you can plug-in an `Online Letter Image` algorithm on the Installation Record. This type of algorithm will call the letter's letter template's extract algorithm to extract the information that is merged onto the letter. Refer to Technical Implementation Of Online Letter Image for the details.
- You can create letters online using an external reporting tool.

Refer to The Doc 1 Records for examples of the record layouts passed to the Doc 1 print software.

- It contains the ID of the background process that builds the flat file that's passed to your letter print software. The base package example of this process (known by the batch control ID of LTRPRT) simply calls each customer contact's letter template's extract algorithm to format the information placed onto the flat file.

BI Publisher and Crystal Reports Versus DOC 1

Using BI Publisher and Crystal Reports as your external reporting tool to display an online letter image differs from using DOC 1 printing software as follows:

- An extract file is not required when using BI Publisher or Crystal Reports. The sample reports provided to use with these external report tools do not have to extract information from another object to merge into a letter.
- Generating letters in batch is not supported yet.

Where Does The Address Come From?

The following points describes the source of a letter's address:

- For letters generated as a result of collection events, write-off events and overdue events, the address is extracted as follows:
  - The respective Account - Person record is retrieved for the letter recipient (note, the account ID is retrieved from the related collection / write-off / overdue process).
  - If the person has a seasonal address effective on the business date, the letter's address is the respective seasonal address.
  - If the address source is `Mailing Location on Account`, the letter's address is the account's mailing location.
  - If the address source is `Account Override`, the letter's address is the override address on the respective Account - Person record.
  - If the address source is `Person`, the letter's address is the *person's mailing address*.
  - If the address source is not defined, the *person's mailing address* is used.

Copyright © 2024, Oracle and/or its affiliates.

  - If an address was not populated as a result of the above, the letter is not produced and a warning is issued by the background process responsible for creating the letter extract file.
- For all other types of letters, the address is extracted as follows:
  - If the letter's person has a seasonal address effective on the business date, the letter's address is the respective seasonal address.
  - If the letter's person has a *mailing address*, it is used.
  - Otherwise, the system checks if the letter's person has at least one account where the person is the main customer and the account has at least one active contract. If so, the logic described above is executed for the first such account found.
  - If an address was not populated as a result of the above, the letter is not produced and a warning is issued by the background process responsible for creating the letter extract file.

Technical Implementation Of Online Letter Images

Users can view an image of any letter that is sent to a customer if you set up the following:

- Plug-in an `Online Letter Image` construction algorithm on the Installation Record. Refer to ONLD-LT for an example of such an algorithm. Note, if your letter print software is not capable of producing a PDF containing an image of a letter, users will not be able to view images of letters.
- Plug-in the appropriate extract algorithm on each letter template. Algorithms of this type format the records that contain the person information that is passed to your printing software.

When you plug-in these algorithms, a button appears on Customer Contact - Main for customer contacts whose customer contact type references a letter template. When a user clicks this button, the following takes place:

- The installation record's `Online Letter Image` construction algorithm is executed.
- This algorithm calls the customer contact's letter template's extract algorithm. This algorithm constructs the information that's merged onto the letter and returns it to the `Online Letter Image` algorithm. This algorithm, in turn, passes it to your letter print software.
- Your letter print software renders an image of the letter in a PDF and returns it to the `Online Letter Image` algorithm.
- And finally, the `Online Letter Image` algorithm displays the PDF in a separate Adobe session.

Technical Implementation Of Batch Letter Production

The batch process that extracts letter information (known by the batch control ID of LTRPRT) reads all customer contact records in a given run number that are marked with its batch control ID. For each customer contact, it creates numerous records on a flat file. These records contain the person information that is merged onto your letters.

The information that is placed on each record is controlled by the customer contact's letter template's extraction algorithm. Refer to Letter Templates Control The Information On Letters for more information.

The records on the flat file can be constructed in either the Doc 1 format (to be used by the Doc 1 software), or in MS Word's field delimited format. The format that's used is controlled by a parameter supplied to the `LTRPRT` background process.

- The Doc1 flat file is created with *multiple* records of various record types for every customer contact record.
- The Word flat file is created with *one* record for every customer contact record processed. Each of the fields is delimited by a special character (e.g. "~").

Note: Alternate print / routing methods. If you require different print formats, you must create new letter extract algorithms and plug them in on your letter template.

Copyright © 2024, Oracle and/or its affiliates.

Reproducing The Letter Print Flat File

You can reproduce the flat file containing the information sent to your printing software at any time. Simply request the respective extract batch process and specify the run number associated with the historic run.

How To Reprint A Specific Letter

If you need to reprint a specific letter, navigate to Customer Contact - Main and turn on the **Reprint** switch (and save the customer contact). Alternatively, if your implementation has enabled the online creation of letter images, you can also click the **Display Letter** button on this page and then print the resultant PDF on your local printer.

The Doc 1 Records

Note: Alternate print / routing methods. This section describes the format of letter extract records that are constructed by base-package algorithms that are passed to the Doc 1 software (which actually produces the letters). If you require different print formats, you must create new letter extract algorithms and plug them in on your letter templates.

Note: Alternate letter print flat file format. The algorithms responsible for creating the letter extract records can produce the output in either Doc1 format (described below) or in Word format (see The Word Records). The Doc1 flat file is created with multiple records of various record types for every customer contact record processed. The Word flat file is created with one record for every customer contact record processed. These algorithms extract the same fields for both formats. However, the Word flat file has all fields for a letter compressed into one record; where each field is delimited by a tilde ("~"). You control which format is used by the parameters you supply to the respective background processes that call these algorithms. For Doc 1 format, set FIELD-DELIM-SW=N and CNTL-REC-SW=Y. For Word format, set FIELD-DELIM-SW=Y and CNTL-REC-SW=N.

Numerous types of records are interfaced to the Doc 1 letter print software. However, they all share a common structure:

- The first 4 bytes are called the **Printer Key**. This field is a record type used by the Doc 1 letter template.
- The next 26 bytes are called the **Sort Key**. This field is used to ensure the various records used to construct a printed letter appear in the correct order in the interface file.
- The next 12 bytes are called the **Mailing Key**.
- The next 3 bytes are a filler that is available for future use.
- The remaining bytes contain the **Letter Information** that appears on the printed letter. The type of information differs for each type of record.

The topics in this section describe each component.

Printer Keys (Record Types)

The following table describes the different record types:

Record Type	Description	Notes
0010	Global extract information	Required. 1 per flat file.
1000	Letter record	Required. 1 per letter.
1100	Mailing Information	Required. 1 per letter.
Note	Record types 1200 through 1600 only exist if the letter was triggered as a result of a collection event, a write-off event, a workflow event or an overdue process event, respectively. These record types are mutually exclusive and it is possible for none to exist (meaning that the letter was triggered by a user-created	

Copyright © 2024, Oracle and/or its affiliates.

Record Type	Description	Notes
	customer contact and therefore has no extract merge information).	
1200	Collection Process/Event	Optional. 0 or 1 record will exist - see Note above.
1400	Write-off Process/Event	Optional. 0 or 1 record will exist - see Note above.
1500	Workflow Process/Event	Optional. 0 or 1 record will exist - see Note above.
1600	Overdue Process/Event	Optional. 0 or 1 record will exist - see Note above.
9990	End of Letter record	Required. 1 per letter (dummy).

Sort Key

The letter print records need to be sorted in the right sequence in order for DOC1 to process them. The following is a description of the portion of each record that is responsible for performing this task.

The following table describes the structure of the sort key that appears on the print record of each letter:

Field Name	Format	Source/Value/Description
LTR_TMPL_CD	A12	
CC_ID	A10	
CC Record Group	A2	This field forces the letter's records into the correct order. Values: 10 - General Letter information 20 - Mailing information 30 - Credit and Collections information 40 - Workflow information 99 - End Letter
C&C Record Group	A2	This field forces the subordinate records into the correct order. Values: 10 - Collection Process/Event information 40 - Write Off Process/Event information 50 - Overdue Process/Event information

Mailing Key

Mailing key contains the postal code.

Letter Information

The topics in this section describe the information that appears on each record type.

Global Information Record

This is a "header" record that contains information about the extract run as a whole.

Copyright © 2024, Oracle and/or its affiliates.

Note: Please be aware that if you do not sort your extract file by **CIS Sort Key** after it is produced, this record will physically be the last record on the extract file. Also note that this record will only be produced by the various letter extract processes if you supply the appropriate parameter value (CNTL-REC-SW=Y). Refer to Extract Processes for more information about the letter extract programs.

The following table lists the header elements that contain information about the extract run:

Field Name	Format	Source/Value/Description
Print Key	A4	"0010"
CIS Sort Key	A26	This is blank for this record type only.
CIS Mailing Key	A15	This is blank for this record type only.
BATCH_CD	A8	CI_BATCH_INST / The CIS Batch Process that called the Letter Extract COBOL routine
BATCH_NBR	N10	CI_BATCH_INST / A sequence number for serial instances of a Batch Process over time
BATCH_RERUN_NBR	N10	CI_BATCH_INST / The number of times a particular batch has been re-run since its initial run
Letter Count [Artificial]	N10	Total number of letters in this extract
Date/Time [Artificial]	A26	System time of extraction.
CC Id Start [Artificial]	A10	The lowest CC ID in a particular thread in the extract
CC Id End [Artificial]	A10	The highest CC ID in a particular thread in the extract
BATCH_THREAD_NBR	N10	CI_BATCH_THD / Identifies a thread distinctly within a group of threads
Thread Count [Artificial]	N10	Total number of threads in the extract run

Letter Record

One record is produced for every letter to be printed.

The following table lists the fields that appear in the letter record:

Field Name	Format	Source/Value/Description
Print Key	A4	"1000"
CIS Sort Key	A26	
CIS Mailing Key	A15	
CC_ID	A10	CI_CC / System-assigned identifier for a Customer Contact.
PER_ID	A10	CI_CC / System-assigned identifier for the Person who will receive the letter.
CC_DT	D10	CI_CC / The date of the Customer Contact.
CC_TM	T15	CI_CC / The time of the Customer Contact.

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
CC_CL_CD	A4	CI_CC / The category of contact.
CC_TYPE_CD	A12	CI_CC / The type of contact within the category.
LTR_TMPL_CD	A12	CI_CC / The template of the letter.
LTR_DT	D10	From the current system date/time - the date to print on the letter.

Mailing Information Record

One record is produced for every letter to be printed.

The following table lists the fields that appear in the mailing record:

Field Name	Format	Source/Value/Description
Print Key	A4	"1100"
CIS Sort Key	A26	
CIS Mailing Key	A15	
ENTITY_NAME1	A64	If an override mailing name has been specified on the CI_PER row, this field contains the first line of the person's override mailing name (OVRD_MAIL_NAME1). Otherwise, the person's primary name is used (ENTITY_NAME).
ENTITY_NAME2	A64	If an override mailing name has been specified on the CI_PER row, this field contains the 2 nd line of the person's override mailing name (OVRD_MAIL_NAME2). Otherwise, this field is blank.
ENTITY_NAME3	A64	If an override mailing name has been specified on the CI_PER row, this field contains the 3 rd line of the person's override mailing name (OVRD_MAIL_NAME3). Otherwise, this field is blank.
ADDRESS_SBR	A361	See below for the fields. Refer to Where Does The Address Come From? for the source of the address information.

Write Off Process / Event Record

This record contains information about the write off process and event that triggered the letter. If the letter was not triggered by a write-off event, this record will not be created. So, there will be at most one write-off event record for a letter.

The following table lists the fields that appear in the write-off event record:

Field Name	Format	Source/Value/Description
Print Key	A4	"1400"
CIS Sort Key	A26	
CIS Mailing Key	A15	
WO_PROC_ID	A10	CI_WO_EVT_CC / Write Off Event that triggered the letter.

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
EVT_SEQ	N3	CI_WO_EVT_CC / Write Off Event that triggered the letter.
ACCT_ID	N10	CI_WO_PROC / Account whose contracts are being written off.
WO_CNTL_CD	C20	CI_WO_PROC / WO control code of the write off process.
WO_DEBT_CL_CD	C4	CI_WO_CNTL / Write Off debt class of the contracts being written off.
DESCR	C30	CI_WO_DEBT_CL.DESCR / Description of the write off debt class.
CRE_DTTM	DT26	CI_WO_PROC / Date/time that the write off process was created.
WO_BALANCE	N13.2	Amount still owed on the contracts linked to the write off process. This is calculated by the same routine that creates an agency referral, (Determine Referral Amount routine (CIPLWRAN)).
CURRENCY_CD	C3	CI_COLL_PROC / Currency for the amount.

Workflow Process / Event Record

This record contains information about the workflow process and event that triggered the letter. If the letter was not triggered by a workflow event, this record will not be created. So, there will be at most one workflow event record for a letter.

The following table lists the fields that appear in the workflow event record:

Field Name	Format	Source/Value/Description
Print Key	A4	"1500"
CIS Sort Key	A26	
CIS Mailing Key	A15	
WF_PROC_ID	A10	CI_WF_EVT / Workflow Event whose event context is the customer contact being processed.
EVT_SEQ	N3	CI_WF_EVT / / Workflow Event whose event context is the customer contact being processed.
WF_PP_CD	A12	CI_WF_PROC / Workflow process profile of the workflow process.
NT_UP_XTYPE_CD	A30	CI_WF_PROC / Notification Upload Type that caused the creation of this process.
DESCR50	A50	CI_NT_UP_XTYPE/ Description of the notification upload type.
CRE_DTTM	DT26	CI_WF_PROC / Date/time that the workflow process was created.
CTXT_VAL_ACCT	A50	CI_WF_PROC_CTXT /Context value where the value of the context flag is ACCT.
CTXT_VAL_MSA	A50	CI_WF_PROC_CTXT /Context value where the value of the context flag is MSA.

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
CTXT_VAL_SPR	A50	CI_WF_PROC_CTXT /CTXT_VAL where the value of the context flag is SPR.
SPR_DESCR50	A50	CI_SPR/DESCR50.
CTXT_VAL_SP	A50	CI_WF_PROC_CTXT /CTXT_VAL where the value of the context flag is SP.
CTXT_VAL_XACC	A50	CI_WF_PROC_CTXT /CTXT_VAL where the value of the context flag is XACC.

Overdue Process / Event Record

This record contains information about the overdue process and event that triggered the letter. If the letter was not triggered by an overdue event, this record will not be created. So, there will be at most one overdue event record for a letter.

Refer to the sample overdue process event extract algorithm for information about the fields below whose source is `Extract Algorithm`.

The following table lists the fields that appear in the overdue event record:

Field Name	Format	Source/Value/Description
Print Key	A4	"1600"
CIS Sort Key	A26	
CIS Mailing Key	A15	
OD_PROC_ID	A10	CI_OD_PROC_LOG / Overdue process that triggered the letter.
OD_EVT_SEQ	N3	CI_OD_PROC_LOG / Overdue event that triggered the letter.
ACCT_ID	N10	CI_OD_PROC / Account being collected on.
OD_PROC_TEMPL_CD	A12	CI_OD_PROC / Overdue process template.
DESCR	C30	CI_OD_PROC_TEMPL / Description of the overdue process template.
CRE_DTTM	DT26	CI_OD_PROC / Date/time that the overdue process was created.
Original Unpaid Amt	N13.2	`Extract Algorithm`/ Amount owed on all objects being collected by the overdue process when the overdue process was created.
Unpaid Amt	N13.2	`Extract Algorithm`/ Amount still owed on all objected being collected by the overdue process.
CURRENCY_CD	C3	CI_OD_PROC / Currency for the amount.
Info String 1		`Extract Algorithm`/ Info about object 1 being collected
Info String 2		`Extract Algorithm`/ Info about object 2 being collected

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
Info String 3		`Extract Algorithm`/ Info about object 3 being collected
Info String 4		`Extract Algorithm`/ Info about object 4 being collected
Info String 5		`Extract Algorithm`/ Info about object 5 being collected

End of Letter Record

The following table lists the fields that appear at the end of the letter:

Field Name	Format	Source/Value/Description
Print Key	A4	"9990"
CIS Sort Key	A26	
CIS Mailing Key	A15	

Definition of Address Subrecord

Used in mailing address and service address, total length = 361 bytes

The following table lists the fields that appear in the address record:

Field Name	Format	Description
COUNTRY	A3	
ADDRESS1	A64	
ADDRESS2	A64	
ADDRESS3	A64	
ADDRESS4	A64	
CITY	A30	
NUM1	A6	
NUM2	A4	
HOUSE_TYPE	A2	
COUNTY	A30	
STATE	A6	
POSTAL	A12	
GEO_CODE	A11	
IN_CITY_LIMIT	A1	

The Word Records

Note: Alternate print / routing methods. This section describes the format of letter extract records that are constructed by base-package algorithms that are passed to Word, the software that actually produces the letters. If you require different print formats, you must create new letter extract algorithms and plug them in on your letter templates.

Copyright © 2024, Oracle and/or its affiliates.

Note: Alternate letter print flat file format. The algorithms responsible for creating the letter extract records can produce the output in either Doc1 format (see The Doc 1 Records) or in Word format (see below). The Doc1 flat file is created with multiple records of various record types for every customer contact record processed. The Word flat file is created with one record for every customer contact record processed. These algorithms extract the same fields for both formats. However, the Word flat file has all fields for a letter compressed into one record; where each field is delimited by a tilde ("~"). You control which format is used by the parameters you supply to the respective background processes that call these algorithms. For Doc 1 format, set FIELD-DELIM-SW=N and CNTL-REC-SW=Y. For Word format, set FIELD-DELIM-SW=Y and CNTL-REC-SW=N.

One record is produced for every letter to be printed. The fields on each record are delimited with a tilde (~).

The following table describes the fields that appear when the extracted record is printed in the Word format:

Field Name	Maximum Size	Source/Value/Description
POSTAL	A12	
ENTITY_NAME1	A64	CI_PER_NAME1 / Primary name of the person on the Customer Contact record.
ENTITY_NAME2	A64	CI_PER_NAME2 / Primary name of the person on the Customer Contact record.
ENTITY_NAME3	A64	CI_PER_NAME3 / Primary name of the person on the Customer Contact record.
COUNTRY	A3	The next group of fields (COUNTRY through IN_CITY_LIMIT) contain the letter's address.
ADDRESS1	A64	
ADDRESS2	A64	
ADDRESS3	A64	
ADDRESS4	A64	
CITY	A30	
NUM1	A6	
NUM2	A4	
HOUSE_TYPE	A2	
COUNTY	A30	
STATE	A6	
POSTAL	A12	
GEO_CODE	A11	
IN_CITY_LIMIT	A1	
CC_ID	A10	CI_CC / System-assigned identifier for a Customer Contact.
PER_ID	A10	CI_CC / System-assigned identifier for the Person who will receive the letter.
CC_DT	See Note	CI_CC / The date of the Customer Contact formatted as <month> DD, YYYY.
CC_TM	T15	CI_CC / The time of the Customer Contact.

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Maximum Size	Source/Value/Description
CC_CL_CD	A4	CI_CC / The category of contact.
CC_TYPE_CD	A12	CI_CC / The type of contact within the category.
LTR_TMPL_CD	A12	CI_CC / The template of the letter.
LTR_DT	See Note	From the current system date/time - the date to print on the letter formatted as <month> DD, YYYY.

Certain types of letters may have additional fields. The topics that follow describe the additional fields that suffix the above fields:

Additional Fields For A Write Off Process / Event Letter

The following table describes additional fields that are suffixed in the letter extract records associated with a write-off event:

Field Name	Format	Source/Value/Description
WO_PROC_ID	A10	CI_WO_EVT_CC / Write Off Event that triggered the letter.
EVT_SEQ	N3	CI_WO_EVT_CC / Write Off Event that triggered the letter.
ACCT_ID	N10	CI_WO_PROC / Account whose contracts are being written off.
WO_CNTL_CD	C20	CI_WO_PROC / WO control code of the write off process.
WO_DEBT_CL_CD	C4	CI_WO_CNTL / Write Off debt class of the contracts being written off.
DESCR	C30	CI_WO_DEBT_CL.DESCR / Description of the write off debt class.
CRE_DTTM	DT26	CI_WO_PROC / Date/time that the write off process was created.
WO_BALANCE	N13.2	Amount still owed on the contracts linked to the write off process. This is calculated by the same routine that creates an agency referral, (Determine Referral Amount routine (CIPLWRAN)).
CURRENCY_CD	C3	CI_COLL_PROC / Currency for the amount.

Additional Fields For A Workflow Process / Event Letter

The following table describes additional fields that are suffixed in the letter extract records associated with a workflow event:

Field Name	Format	Source/Value/Description
WF_PROC_ID	A10	CI_WF_EVT / Workflow Event whose event context is the customer contact being processed.
EVT_SEQ	N3	CI_WF_EVT / / Workflow Event whose event context is the customer contact being processed.
WF_PP_CD	A12	CI_WF_PROC / Workflow process profile of the workflow process.
NT_UP_XTYPE_CD	A30	CI_WF_PROC / Notification Upload Type that caused the creation of this process.
DESCR50	A50	CI_NT_UP_XTYPE/ Description of the notification upload type.
CRE_DTTM	DT26	CI_WF_PROC / Date/time that the workflow process was created.

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
CTXT_VAL_ACCT	A50	CI_WF_PROC_CTXT /Context value where the value of the context flag is ACCT.
CTXT_VAL_MSA	A50	CI_WF_PROC_CTXT /Context value where the value of the context flag is MSA.
CTXT_VAL_SPR	A50	CI_WF_PROC_CTXT /CTXT_VAL where the value of the context flag is SPR.
SPR_DESCR50	A50	CI_SPR/DESCR50
CTXT_VAL_SP	A50	CI_WF_PROC_CTXT /CTXT_VAL where the value of the context flag is SP.
CTXT_VAL_XACC	A50	CI_WF_PROC_CTXT /CTXT_VAL where the value of the context flag is XACC.

Additional Fields For An Overdue Process / Event Letter

The following table describes additional fields that are suffixed in the letter extract records associated with an overdue event:

Field Name	Format	Source/Value/Description
OD_PROC_ID	A10	CI_OD_PROC_LOG / Overdue process that triggered the letter.
OD_EVT_SEQ	N3	CI_OD_PROC_LOG / Overdue event that triggered the letter.
ACCT_ID	N10	CI_OD_PROC / Account being collected on.
OD_PROC_TEMPL_CD	A12	CI_OD_PROC / Overdue process template.
DESCR	C30	CI_OD_PROC_TEMPL / Description of the overdue process template.
CRE_DTTM	See Note.	CI_OD_PROC / Date/time that the overdue process was created formatted as <month> DD, YYYY.
Original Unpaid Amt	N13.2	`Extract Algorithm`/ Amount owed on all objects being collected by the overdue process when the overdue process was created.
Unpaid Amt	N13.2	`Extract Algorithm`/ Amount still owed on all objected being collected by the overdue process.
CURRENCY_CD	C3	CI_OD_PROC / Currency for the amount.
Info String 1		`Extract Algorithm`/ Info about object 1 being collected
Info String 2		`Extract Algorithm`/ Info about object 2 being collected
Info String 3		`Extract Algorithm`/ Info about object 3 being collected
Info String 4		`Extract Algorithm`/ Info about object 4 being collected
Info String 5		`Extract Algorithm`/ Info about object 5 being collected

Maintaining Declarations

The declaration page allows you to define a declaration percentage for a combination of account, service type, declaration type and date range. The declaration may be used by algorithms to adjust certain calculations in the system. For example

Copyright © 2024, Oracle and/or its affiliates.

you may give certain customers a reduced VAT rate. The declaration is used to indicate the reduced rate for the qualifying customer.

Open **Customer Management, Declaration** to add or update a declaration.

> **Note: Applicable for UK market**. This page is intended for the United Kingdom market. Other markets may disregard this transaction.

Description of Page

The **Declaration Info** is a concatenation of important information about the declaration record. The **Declaration ID** is a system-assigned random number that uniquely identifies this record.

Use **Account ID** to indicate the account for this declaration.

Use **Location ID** to indicate the location for this declaration.

The **Service Type** controls the contracts covered by this declaration. All contracts for this account for this service type are covered.

Use the **Declaration Type** to indicate the type of declaration. The system includes the declaration types `CCL Relief` and `Reduced Rate VAT`.

> **Note:** The values for this field are customizable using the Lookup table. This field name is DCL_TYPE_FLG.

Enter the **Start Date** and **End Date** to indicate the effective period of this declaration. The start date defaults to the current date.

Enter the **Declaration Percentage**.

If this declaration is related to an external document, indicate the **Document Reference**.

This bottom of this page contains a tree that shows information about all the declarations linked to this account. You can use this tree to both view high-level information about these objects and to transfer to the respective page in which an object is maintained.

Chapter 2

Billing

Topics:

- The Big Picture of Billing
- Maintaining Bills
- Financial Transactions On A Bill
- Maintaining Bill Segments
- Multi Cancel/Rebill
- Contract Billing History
- Bill Exception
- Bill Segment Exception
- Invoice Frequency
- Charge Type
- Maintaining Billable Charges
- Uploading Billable Charges
- Budget Review
- Bill Segment Summary
- Bill Drill Down Details
- Alert Notification for Bill and Bill Segment Errors

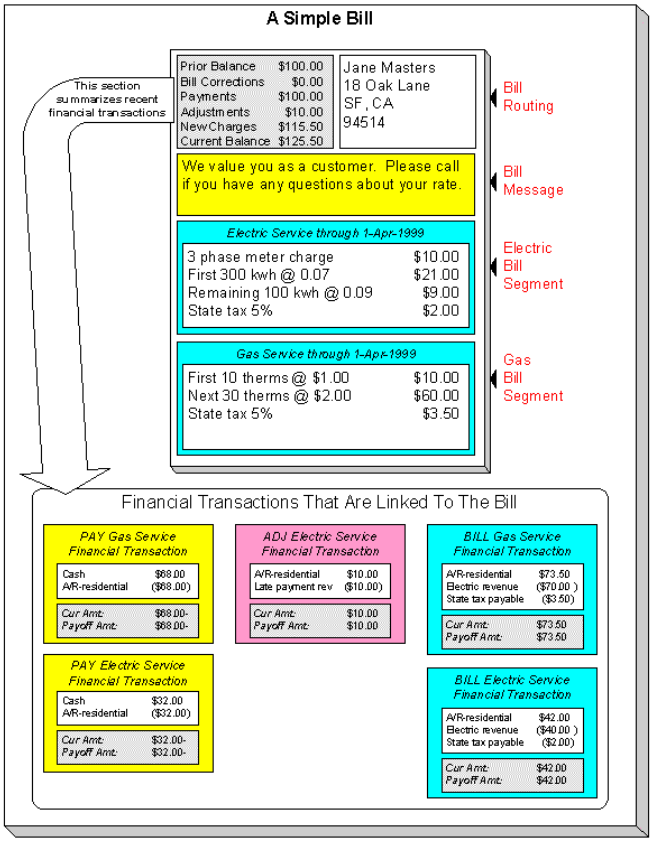
In this section, we describe how to manage your customer's bills.

The Big Picture of Billing

A bill is used to communicate changes in an account's financial obligations to the customer. Over time, a customer receives many bills. The topics in this section provide background information about a variety of billing topics.

An Illustration Of A Simple Bill

The following diagram illustrates a simple bill:



The following concepts are illustrated above:

A bill is produced for an account A bill is produced for an account. Over time, an account receives many bills.

A bill summarizes financial transactions A bill contains information about the various financial transactions that have taken place since the last bill was produced (i.e., payments, adjustments, and bill corrections). The above illustration shows a bill with financial transactions for new charges, a payment, and an adjustment.

A bill is routed to persons A copy of the bill is sent to every person linked to the account who requires a copy of the bill.

A bill contains messages A bill may contain messages.

A bill typically contains bill segments A bill typically contains one bill segment for every active contract linked to its account.

A bill segment contains calculation details A bill segment contains information showing how the segment was calculated and how it should be printed on the customer's bill.

Copyright © 2024, Oracle and/or its affiliates.

A High Level Overview Of The Bill Creation Process

When the system is asked to produce a bill for an account, it attempts to create one or more bill segments for every non-cancelled / non-closed contract linked to the account. Whether or not a contract contributes bill segment(s) to a bill is a complicated subject as the system supports a wide variety of bill segment creation methods.

The system applies the contract's rate to the amount consumed in order to calculate how much the customer owes. A bill segment and a financial transaction are generated to reflect the results of this calculation. The details of the calculations are captured in the bill calculation lines. For more information, refer to How Rates Affect The Information On Bill Segments for the details. Refer to Effective Dates & Price Proration for information about how the system prorates changes to rates and prices during a bill period.

Note: There are many bill segment creation algorithms. The details given above explain how a bill segment is produced for a contract. It's important to be aware that the system has other bill segment creation algorithms. For example, if the contract charges for a deposit, the system neither amalgamates consumption nor does it apply a rate. You may be wondering how the system knows which method to use. It uses the bill segment type on the contract's contract type. Refer to Designing and Defining Bill Segment Types for a description of the various bill segment creation algorithms that are supported in the system.

If errors are detected during the bill segment creation process, the bill segment is saved with its error. The system then proceeds to the account's next contract. This way, a user can see all problematic bill segments so they can be corrected en masse. Refer to Bill Errors for information about how to deal with bill errors.

Note: Interdependent contracts. Your organization may have some contracts whose charges must be calculated in a specific order. For example, you might have a contract whose bill segments are calculated using charges from a different bill segment. If this describes your organization, be aware that the system creates the bill segments in the order of the billing processing sequence on each contract's contract type.

When every segment on a bill is error free, the bill is ready to be completed. When a bill is completed, the system:

- Creates a bill routing for each person linked to the account who requires a copy of the bill. The routing information controls the format of the printed bill and how the bill is sent to the person. Refer to The Source Of Bill Routing Information for the details.
- Links all bill messages to the bill. A bill message can come from a variety of sources. Refer to The Source Of Bill Messages for the details.
- If the `Freeze At Bill Completion` installation option has been turned on:
  - All `freezable` bill segments will be frozen.
  - All `freezable` adjustments whose adjustment type indicates `Freeze At Bill Completion` will be frozen.
- Sweeps recent financial transactions that have been created since the last bill was completed. Refer to Financial Transactions Created Between Bills for information about how payments, adjustments and bill corrections appear on a bill.
- There are several other functions that happen at completion time. Refer to the description of the Complete button under Bill Lifecycle for the details.

And that's it; the bill can now be sent to the customer.

The remaining topics in this section provide more information about the creation and completion of bills.

Note: Batch and real-time bill creation. Anything the batch bill process does for whole sets of accounts, you can do to a specific account on-line / real time. Refer to How To Create A Bill For All Contracts Linked To An Account for information about how to create a bill on-line / real time. Refer to Batch Billing for more information about the batch bill creation process.

Copyright © 2024, Oracle and/or its affiliates.

Bill Errors

The topics in this section describe errors that are detected when the system creates a bill.

Bill Segment Errors

If you consider the huge number of variables involved in the creation of a bill segment, it probably won't surprise you to learn that occasionally, some bill segments are in error.

Errors tend to be caused by missing or inconsistent data. Some examples of classic errors:

- **Missing master data.** For example, if a bill is supposed to be routed to the account's mailing address and the account doesn't have such an address, an error is generated.
- **Missing rate data.** For example, if a rate contains a customer specific charge and there is no value defined on the customer's contract, an error is generated.

The system saves bill segments that are in error just as it saves bill segments that are error-free. This is done because bill segments are nothing more than a snapshot of the data that was used to calculate the charges. By saving the snapshot, you can see the information the system used when it detected the error and therefore more effectively fix the cause of the error. The standard way to fix an error is to:

- Look at the bill segment to determine the cause of the error.
- Correct the cause of the error.
- Regenerate the bill segment. Regeneration simply deletes the offending segment and recreates a new segment using the corrected information.

For every bill segment in error, a record is written to the Bill Segment Exception table.

Fastpath: Refer to How To Correct A Bill Segment That's In Error for instructions describing how to correct a bill segment.

It's obvious but worth stressing that a bill may contain some segments that are perfect and others that are in error. Once all segments that are in error are corrected, the bill can be completed (and sent to the customer).

If the bill segment in error was created as part of the batch bill process, the system attempts to fix the offending segment(s) each night during the account's bill window by regenerating it. Therefore, if the cause of the error is fixed during the day, the system will automatically regenerate the bill segment when batch billing next runs; you don't have to manually correct each bill. And once a bill is error-free, it will be completed and sent to the customer.

Fastpath: Refer to Batch Billing for more information.

Bill Completion Errors

In addition to errors on bill segments, there may also be errors detected when the system attempts to complete a bill. For example, if the system cannot find a mailing address, the bill will be in error (as opposed to one of its bill segments).

Errors tend to be caused by missing or inconsistent data. Some examples of classic errors:

- **Missing mailing address.** For example, if the system cannot find an address to which the bill can be routed, it generates an error.
- **Bill segments are in error.** If a bill contains bill segments that are in error, it cannot be completed and an error is generated.

The system saves bills that are in error just as it saves bills that are error-free. This is done because bills are nothing more than a snapshot of the data that was used to calculate the charges. By saving the snapshot, you can see the information the system used when it detected the error and therefore more effectively fix the cause of the error. The standard way to fix an error is to:

- Look at the bill to determine the cause of the error.
- Correct the cause of the error.

Copyright © 2024, Oracle and/or its affiliates.

- Recomplete the bill.

For every bill in error, a record is written to the Bill Exception table.

Fastpath: Refer to How To Correct A Bill That's In Error for more information.

If the bill completion error was created as part of the batch bill process, the system attempts to recomplete the bill each night during the account's bill window. Therefore, if the cause of the error is fixed during the day, the system will automatically complete the bill when batch billing next runs; you don't have to manually complete each bill.

Fastpath: Refer to Batch Billing for more information.

Cancel / Rebill Incorrect Bill Segments

Sometimes the error on a bill segment is not detected by the system. Such errors occur when the data used to calculate the bill is valid, but wrong. For example, the sales tax percent was entered incorrectly on a bill factor.

There is no way the system can detect such problems and therefore the system freezes the bill segment and routes the bill to the customer. To correct such a bill segment, you must cancel the offending segment and create a new segment (after correcting the cause of the problem). We refer to this process as cancel / rebill.

Fastpath: Refer to How To Cancel / Rebill A Bill Segment for more information about how to cancel / rebill a frozen bill segment.

You must use the cancel / rebill process to correct problems on frozen bill segments. This is because a frozen bill segment can be thought of as having been posted to your general ledger (even if the GL interface hasn't run). And once a financial transaction is posted to the general ledger, it can only be removed via a reversal. Contrast this to bill segments that are in error - they can be deleted because they were never posted to the general ledger.

Fastpath: Refer to Bill Segment Lifecycle for more information about the differences between bill segments that are frozen and those that are in error.

Note: Bill segments are canceled / rebilled. It's important to stress that bill segments are canceled and rebilled (as opposed to bills). If every bill segment on a bill is incorrect, you must cancel / rebill each individual bill segment.

Note: Using the cancel or cancel/rebill functionality will cause the cancel and rebill details to be swept on to the customer's next bill. If your organization requires cancellation details to be presented to the customer separately from the new bill, then you will need to use Credit Notes. Refer to Credit Notes - Presenting Cancellation Details on a Separate Bill for more information. If your organization requires cancellation and rebill details to be presented to the customer separately from the new bill, then you will need to use Correction Notes. Refer to Correction Notes - Presenting Correction Details on a Separate Bill for more information.

Credit Notes - Presenting Cancellation Details on a Separate Bill

The system's Cancel / Rebill functionality will include cancellation and rebilling details on the customer's next bill. Some organizations are required to present cancellation details to the customer separately and not to commingle them with a new bill. This separate statement containing only the cancellation information is referred to as a Credit Note.

Credit Notes may be created for an individual bill segment or for all the bill segments linked to a bill. To create a credit note, a Credit Note button is available on the Bill Segment page and on the Bill page.

Fastpath: Refer to How To Create A Credit Note for more information about how to create credit notes for a bill or bill segment.

The topics in this section provide describe the impact of this functionality.

Copyright © 2024, Oracle and/or its affiliates.

Credit Notes Are Only Possible if the Installation Option Is Turned On

The Credit Notes button on the bill segment and bill pages will only be available if you have selected the Credit Note Bill Correction option on the Installation record.

Credit / Correction Notes and Auto Pay

If a credit or correction note is being created for a customer who is enrolled in Auto Pay, note the following functionality:

When creating a credit or correction note for a bill (and all its bill segments):

- The system will cancel an automatic payment behind-the-scenes if a credit or correction note is issued BEFORE the automatic payment is interfaced to the financial institution. When you recomplete the bill, the system will create a new automatic payment that reflects the new amount due (and the canceled automatic payment will net out the original automatic payment). This is standard auto pay logic. Refer to How Are Automatic Payments Cancelled? for more information.
- If the customer has automatic payments that have already been interfaced to the financial institution, the creation of the credit or correction note will have no affect on any automatic payment. A warning will be issued in this case.

When creating a credit or correction note for a bill segment:

- Because automatic payments are for an entire bill, creating a credit or correction note at the bill segment level will not affect any automatic payments. A warning is issued in this case.

Fastpath: For more information about correction notes, see Correction Notes - Presenting Correction Details on a Separate Bill.

Adjustments On Credit Notes

Above, we described how a credit note is a separate document that contains the cancellation details of the various bill segments that appeared on a given bill. Some implementations also want the adjustments that appeared on the original bill to be cancelled when a credit note is produced. Then, when the next bill is produced for the customer, they want these same adjustments reproduced. The following points describe how to implement this functionality:

- Plug in a `Bill Completion` algorithm on the applicable Customer Classes. A base-package algorithm (C1-CN-RADJ) will reverse a bill's adjustments when a credit note is completed.
- Plug in a `Post Bill Completion` algorithm on the applicable Customer Classes. A base-package algorithm (C1-CN-ADNB) will recreate the adjustments on the original bill after the credit note is completed. These adjustments will be swept onto the next bill produced for the account.

You can restrict the above functionality to specific adjustment types (for whatever reason). You do this by putting a given char type / char value on the adjustment types that should be reversed and recreated when a credit note is created. You define this char type / char value combination as parameters of the two algorithms described above.

Note: When the original bill contains adjustment cancellations. The above discussion explained how adjustments can be canceled on a credit note and then recreated on the next bill. These same algorithms also do the opposite, i.e., if the original bill contains an adjustment cancellation, the credit note will contain the recreated adjustment, and the next bill will contain its cancellation. Without this, the credit note would not be the reversal of the bill (and this is how some organizations operate).

Credit Note Routing Suppression

Some implementations do not send credit notes to their customers. To implement this functionality, plug in a `Bill Completion` algorithm on the applicable Customer Classes. A base-package algorithm (C1-CN-BRT) will cause the routing information to be suppressed when a credit note is completed.

Copyright © 2024, Oracle and/or its affiliates.

Correction Notes - Presenting Correction Details on a Separate Bill

The system's Cancel / Rebill functionality will include cancellation and rebilling details on the customer's next bill. Some organizations are required to present cancellation and rebilling details to the customer separately and not to commingle them with a new bill. This separate statement containing the cancellation and rebill information is referred to as a Correction Note.

Correction notes may be created for an individual bill segment or for all the bill segments linked to a bill. To create a correction note, a **Correction Note** button is available on the Bill Segment and Bill pages.

Fastpath: Refer to How To Create A Correction Note for more information about how to create correction notes for a bill or bill segment.

The topics in this section describe correction notes.

Correction Notes Are Only Possible if the Installation Option Is Turned On

The **Correction Note** button on the bill segment and bill pages will only be available if you have selected the Correction Note Bill Correction option on the Installation record.

Fastpath: If your implementation uses `Correction Notes`, it is recommended to also use the `Freeze At Bill Completion` installation option. For more information, see Installation Options - Billing.

Correction Notes Versus Credit Notes

Correction notes are a mixture of Credit Note and Cancel / Rebill functionality. However, unlike credit notes, correction notes are like regular bills and are subject to payment.

When you issue a correction note for a bill, the system cancels and rebills the bill segments for that bill. The new bill (the correction note) contains both the cancelled bill segments and the rebilled bill segments.

Similar to credit notes, the new bill (or correction note) is stamped with the ID of the original bill. After you review the bill segments, you can complete the correction note. During completion, the rebilled bill segments and the cancelled financial transactions (of the original bill's bill segments) are swept onto the correction note.

Correction Notes and Auto Pay

Fastpath: Refer to Credit / Correction Notes and Auto Pay for a description of how the system handles correction notes for customers using auto pay.

How To Override The Commodity Amount In A Bill Segment

As described earlier, the calculation of the amount consumed is one of the first steps that takes place when a bill segment is created. 99.9% of the time, the system does a perfect job in this effort. However, there are rare circumstances when, due to missing or incorrect data, the quantity calculated by the system is not what you want to charge the customer.

To help you deal with these rare situations, a "back door" exists that allows a bill segment's billable quantities to be overridden. This truly is a "back door" because the proper way to fix the amount of a commodity is to correct the cause of the error and then cancel / rebill the bill segment.

Fastpath: Refer to How To Override Service Quantities On A Specific Bill Segment for more information.

Copyright © 2024, Oracle and/or its affiliates.

How Rates Affect The Information On Bill Segments

After product's consumption has been compiled, the system applies the contract's rate to the product's consumption to determine how much to charge the customer. We refer to this process as "rate application". Refer to Rates for detailed information about rates.

Note: Not all contracts use a rate! There are a small number of contracts that don't use a rate to calculate the charges that appear on their bill segments. For example, a contract used to levy a customer's payment arrangement amount doesn't need a rate because the amount to charge is defined on the contract. Whether or not a contract needs a rate is defined by the contract's contract type. Whether or not the system calls the contract's rate to calculate charges is controlled by the bill calculation algorithm plugged-in on the bill segment type that's referenced on the contract's contract type. Refer to Designing and Defining Bill Segment Types for a description of the various bill segment creation algorithms that are supported in the system.

Note: Your proration choices impact what appears on a bill. A rate contains a variety of effective-dated information (e.g., the prices are effective-dated, the structure of the rate is effective-dated, etc.). If this effective-dated information changes during a bill period, the system may need to prorate the charges. For example, if the sales tax percentage changes mid-period, the system can prorate the tax change (e.g., 20 days at 6% and 11 days at 6.25%). When you setup a rate, you define exactly how the system handles changes that occur during a bill segment. For example, you can tell the system that sales tax changes should not be prorated. Rather, it should use the value effective at the start / end of the bill period (note, you have several other options). We mention this because your choices have a large impact on how a rate affects the information on bill segments. Refer to Effective Dates & Price Proration for more information.

Rate application is a very sophisticated process as it can affect every aspect of a bill segment. The following points describe how rate application works at a high-level:

- The read details that were amassed earlier are passed into rate application.
- Next, the total amount of each unique unit of measure / Variance Parameter that's referenced on every read detail is summarized in the bill segment's "service quantity (SQ) details".
- Next, the *rate schedule's service quantity rules* (if any) are executed. Service quantity rules are used to calculate service quantities (SQ) referenced in the rate that cannot be derived from the read details. For example, if you have a rate that has a charge based on the number of days in the bill period, a service quantity rule would be necessary to calculate the number of days in the bill period.
- Next, the system determines which of the rate's rate versions should be processed. Note, a flag on the rate schedule allows you to control what happens if multiple rate versions are detected during the bill period. A bill segment "calculation detail" is created for each such rate version. Each calculation detail contains the results of executing the rate version's rate components (where a separate bill line is created for each rate component). Each bill line contains:
  - An audit of what was priced.
  - How the printed bill line should look (if the bill lines is printed).
  - How the amount of the bill line should be booked in the general ledger.

Note: There are many plug-in spots available on a rate component. These plug-ins can manipulate virtually every aspect of the bill segment. This means if you require functionality that isn't support by the base package rate components, you can build additional rate component plug-ins to do whatever you need.

- After every rate version is processed, rate application returns the calculation details to the process that called it. This point is important as it means that rate application executes in memory (i.e., it does NOT insert the calculation details on the database, rather, it returns them to whatever process called rate application). Because rate application executes in memory, it can be used to perform billing calculations in many parts of the system. For example:
  - Both online and batch billing use rate application to calculate the charges that are eventually saved on the bill's bill segments.
  - The Rate Check transaction calls rate application when you want to check a rate real-time. When you use this transaction, you enter the product's consumption that's passed into rate application and then rate application shows you the calculation details that result.

Copyright © 2024, Oracle and/or its affiliates.

- Plug-ins exist that call rate application to estimate bill segments when adequate billing history doesn't exist. For example, the BUDCALC-PH budget recommendation algorithm calls rate application to calculate a budget amount for a customer without adequate history. Another example is the DEPRECOM-MBT deposit recommendation algorithm. This plug-in calls rate application to calculate a recommended deposit amount for a customer without adequate history.

Fastpath: Refer to Rates for more information about how rates are constructed.

Bill Frequency - Bill Cycle vs Bill Segment Duration

An account's bill cycle defines when the system *attempts* to create bill segments for the account's contracts. The word "attempt" is stressed because a contract may not have a bill segment on every bill created for its account.

Some examples will help make the point:

- An account may be on a monthly bill cycle (meaning we attempt to create a bill every month for the account) but contain only biannual contracts. However, this is not sensible. Why? Because the system would attempt to create a bill every month for the account's contracts, but only twice during the year would it succeed (because the contracts have a biannual duration).
- An account may be on a biweekly bill cycle (meaning we attempt to create a bill every 2 weeks for the account) and contain a mixture of biweekly, monthly, and quarterly contracts. In this scenario, every two weeks the system would create a bill that contains at least one bill segment (for the biweekly contract). However, 12 times a year, the bill will contain an additional bill segment for the monthly contract. And 4 times a year, the bill will contain one more bill segment (for a total of 3) for the quarterly contract.

In sum, the account's bill cycle controls when the system attempts to create a bill for the account's contracts. Whether a contract contributes a bill segment to the bill is a complicated subject. The topics in this section describe how the system knows it's time to create a bill segment for a contract.

Note: Important! An account's bill cycle should attempt to create bill segments at least as often as the shortest contract duration. For example, if an account has both monthly and quarterly contracts, the account should be placed on a monthly bill cycle. Refer to How Does An Account Get Its Bill Cycle? for more information.

Fastpath: For more information about bill cycles, refer to Defining Bill Cycles.

Ways To Control The Start Date Of A Bill Segment

Bill segments produced for a contract have two time periods:

- The bill segment period. The bill segment period defines the entire period of time covered by a bill segment's charges.
- The consumption period. The consumption period defines the period of time used to calculate the number of days for daily charges.

The consumption period almost always starts one day after the bill segment period. The consumption period always ends on the bill segment's end date. For example, a bill segment period that spans 5-Jan-2002 through 6-Feb-2002 will almost always have a corresponding consumption period of 6-Jan-2002 through 6-Feb-2002. The reason that the start dates don't match is because a bill segment's start date equals the end date of the prior bill segment (i.e., the start date was already counted in the previous bill segment's consumption period and we don't want to count it twice).

The only time when the previous paragraph isn't true is the first bill segment that's produced for a new contract. This is because different companies count the first day of a new contract differently than others.

Ways To Control The End Date Of A Bill Segment

The following points describe the different methods that can be used to define the end date on a contract's bill segments:

Copyright © 2024, Oracle and/or its affiliates.

- **Bill Period Schedule.** Contracts may have the end date of their bill segments defined on the user-maintained bill period calendar. This option is used when bill segments must fall on strict calendar boundaries (e.g., quarterly bills that end on the last day of the quarter).
- **Billable Charge.** For contracts that exist to levy billable charges, the start AND end dates are defined on the billable charge information that is entered by a user or interfaced from an external system.

The topics in this section discuss each of the above methods.

Using The Bill Period Schedule Method

The bill period schedule method causes the bill segment end date to be calculated using a bill period schedule. The bill period's schedule defines when bill segments are produced and their respective end dates. This method is used when the bill segment needs to fall on strict calendar boundaries (e.g., quarterly bills that end on the last day of the quarter).

The contract's contract type indicates if the contract uses this method, and if so, the specific bill period whose schedule defines the end dates.

Note: Future and Past End Dates. It's important to be aware that a bill period schedule can be used to generate end dates that are in the future or in the past.

Fastpath: For more information about bill periods, refer to Designing Bill Periods.

Using The Anniversary Method

The anniversary date method causes the bill segment end date to be calculated based on the first day of service. For example, if a contract started on the 16 th of some month, the ongoing bill segments will start on roughly the 16 th of each month. This method is frequently used to bill for garbage or cable service. When using this method, you also define the applicable billing frequency (i.e., monthly, bi-monthly, quarterly, etc.).

The following table contains the bill periods for a contract's starting on 23-Feb-99 that uses a monthly frequency.

Start Date	End Date	Billing Days
23-Feb-99	25-Mar-99	30
25-Mar-99	25-Apr-99	31
25-Apr-99	25-May-99	30
25-May-99	25-Jun-99	31
25-Jun-99	25-Jul-99	30
25-Jul-99	25-Aug-99	31
25-Aug-99	24-Sep-99	30
24-Sep-99	24-Oct-99	30
24-Oct-99	24-Nov-99	31
24-Nov-99	24-Dec-99	30
24-Dec-99	24-Jan-00	31
24-Jan-00	23-Feb-00	30
23-Feb-00	25-Mar-00	31

Note: Future and Past End Dates. It's important to be aware that anniversary billing can generate end dates that are in the future or in the past. The future-option generates the next anniversary date after the current date. The past-option generates the anniversary date that immediately precedes the current date.

The contract's contract type indicates if the contract uses this method, and if so, the billing frequency.

Copyright © 2024, Oracle and/or its affiliates.

> **Fastpath:** For more information about the contract type attributes that control this method, refer to Setting Up Contract Types.

Using Billable Charges

Bill segments for billable charge contracts have both their start and end dates defined on the respective billable charge.

> **Fastpath:** For more information about billing billable charges, refer to How To Create A One-Time Invoice. For more information about creating a billable charge, refer to Maintaining Billable Charges. For more information about interfacing a billable charge from an external system, refer to Uploading Billable Charges.

The contract's contract type indicates if the contract uses this method, and if so, the billing frequency.

> **Fastpath:** For more information about the contract type attributes that control this method, refer to Setting Up Contract Types.

Preventing Short Bill Segments

Every contract type contains an attribute defining the minimum number of days on a bill segment. Whenever the system attempts to create a bill segment other than the final bill segment, it checks if the number of days is at least as great as the minimum. If not, the bill segment will not be created as part of this bill run. Rather, the system waits until the number of days in the bill segment is at least as large as the minimum. This is true for all methods of bill duration calculation.

Prorating Charges When a Rate is Applied

It is possible for some of the prices that appear on a bill segment to change during the course of the bill period.

> **Fastpath:** Refer to Effective Dates & Price Proration for more information about how charges are prorated during rate application.

Batch Billing

99.9% of all bills are created by the batch bill process (known by the batch control ID of `BILLING`) and require no human intervention before they can be finalized and sent to a customer. The other 0.1% are created by users on-line / real time. This section discusses several important concepts associated with batch billing.

Window Billing And The Bill Cycle Schedule

> **Fastpath:** Refer to The Cyclical Billing Process & Window Billing for more information about how an account's bill cycle dictates when a customer is billed.

Confirming A Batch Of Bills Before Completing Them

If you're implementing new rates or if something unusual is being introduced that affects billing, you may want to turn off the bill cycle schedule's Freeze and Complete switch. If this switch is off, the system creates bills, it just doesn't freeze and complete them. You can then review the entire batch of bills to make sure they're clean.

- If you find the bills are wrong, correct the source of the error and rerun the bill cycle. The system will remove all incomplete bills and then reproduce them using the corrected information in the system.
- If you find the bills are correct, turn the Freeze and Complete switch on and rerun the cycle. The system will remove all incomplete bills and then reproduce them. Assuming nothing changed, the bill amounts will be the same.

> **Fastpath:** Refer to Setting Up Bill Cycle Schedules for more information.

Copyright © 2024, Oracle and/or its affiliates.

Canceling A Batch Of Bills After They're Complete

If you need to cancel an entire batch of bills because they were created using faulty data (e.g., the wrong tax rate was defined in the rate), you can. A background process called MASSCNCL exists for this purpose.

This background process will cancel all the frozen bill segments for the latest run of a given bill cycle's schedule. Optionally, you can cancel bills for a given bill date within the bill cycle's schedule. The cancel reason used on the bill segment is the one marked as the bill cancel reason for mass cancel. Refer to Setting Up Bill (Segment) Cancellation Reasons for more information.

When the cycle is billed again, new bill segments will be created, and the original bill segments and the cancellations will automatically be hidden from the customers.

Reopening A Batch Of Bills After They're Complete

If you need to reopen an entire batch of bills because they were completed prematurely (e.g., payments did not get posted before billing ran, so recent payments were not reflected in the account balances), you can. A background process called MASSROBL exists for this purpose.

This background process will reopen all the bills for a given bill date for the latest run of a given bill cycle's schedule.

Fixing Errors Detected In Batch Billing

If an "error" bill segment is created by the batch billing process, the system attempts to fix the offending segment each night by regenerating it using the information that exists at that time. Therefore, if the cause of the error is fixed during the day, the system will automatically regenerate the bill segment when batch billing next runs; you don't have to manually correct each bill. And, once a bill is error-free, it will be completed and sent to the customer.

Note: Important! Automatic regeneration only works during the account's bill window. If an "error" bill segment is not successfully regenerated on the last night of the account's bill cycle, it will remain in error until the next time the account's cycle runs (unless a user corrects it real time). At that time, the system will generate another bill error indicating the prior bill segment is in error (and then there'll be two bill segments in error).

Completing Pending Bills

If you need to complete pending bills created by the batch billing process, you can. A background process called C1-BLCMP, exists for this purpose.

This background process will complete all pending bills on a given bill date for a given bill cycle's schedule. This process does not delete and regenerate freezable bill segments linked to pending bills.

Fastpath: Refer to Bill Errors for more information.

Billing Financial Transaction Considerations

The topics in this section provide information about the financial impact of a bill segment.

Billing - Current Balance versus Payoff Balance

A bill segment's financial transaction affects a contract's payoff balance and/or current balance. In this section, we describe these two balances.

CAUTION: If you do not understand the difference between payoff balance and current balance, refer to Current Amount versus Payoff Amount.

When Current Balance Equals Payoff Balance

For most contracts, payoff balance and current balance are always the same (or in colloquial speech - the amount the customer thinks they owe equals what they really owe). Let's run through a typical example. The values in the payoff

Copyright © 2024, Oracle and/or its affiliates.

balance and current balance columns reflect the amount due after the financial transaction has been applied (i.e., the running balance):

Date	Financial Transaction	Payoff Balance	Current Balance
1-Jan-99	Bill: $125	125	125
15-Jan-99	Payment: $150	-25	-25
2-Feb-99	Bill: $175	150	150
14-Feb-99	Payment: $150	0	0
3-Mar-99	Bill: $200	200	200
15-Mar-99	Payment: $150	50	50
2-Apr-99	Bill: $225	275	275

As you can see, payoff balance and current balance are always in sync.

When Current Balance Differs From Payoff Balance

For some contracts, payoff balance and current balance differ (or in colloquial speech â€‘ the amount the customer thinks they owe differs from what they would owe if they wanted to pay off their account).

Let's run through an example of a customer on a budget to illustrate a classic contract where these two balances are not the same. The values in the payoff balance and current balance columns reflect the amount due after the financial transaction has been applied:

Date	Financial Transaction	Payoff Balance	Current Balance
1-Jan-99	Bill: $125, Budget $150	125	150
15-Jan-99	Payment: $150	-25	0
2-Feb-99	Bill: $175, Budget $150	150	150
14-Feb-99	Payment: $150	0	0
3-Mar-99	Bill: $200, Budget $150	200	150
15-Mar-99	Payment: $150	50	0

The following table describes the various types of contracts where these balances may differ.

Type Of Contract	What Payoff Balance Holds	What Current Balance Holds
Deposit	Payoff balance holds the amount of deposit you are holding on behalf of the customer's debt. This amount is showed as a credit (with a negative sign) because you truly owe the customer this money.	Current balance holds the amount the customer owes for a deposit.
Charitable contribution	N/A - charitable contribution contracts never have a payoff balance.	Current balance holds the amount the customer owes for a charitable contribution.
Payment arrangements	Payoff balance holds the total amount of delinquent debt that remains unpaid. When a payment arrangement is started, you transfer the payoff balance from the financial service	Current balance holds the amount the customer owes in respect of the billed payment arrangement amounts.

Copyright © 2024, Oracle and/or its affiliates.

Type Of Contract	What Payoff Balance Holds	What Current Balance Holds
	contract to the payment arrangement contract. Each billing period, the customer pays down this balance.	
Contracts on a budget plan	Payoff balance holds the actual amount owed (actual bills minus payments).	Current balance holds the amount the customer owes in respect of the billed budget amount.

Bill Segment Type Controls Which Balance(s) Are Affected

Every bill segment references a bill segment type (the bill segment type comes from its contract's contract type). The bill segment type controls how payoff balance and current balance are affected by the bill segment amount. It also controls the algorithm used by the system to calculate the bill segment's bill lines.

Fastpath: Refer to Designing and Defining Bill Segment Types for more information about how bill segment type affects how a bill segment is produced and how its financial transaction is generated.

The Source Of GL Accounts On A Bill Segment's Financial Transaction

A bill segment's financial transaction also contains the double-sided accounting entry that defines how the bill segment affects the general ledger.

Fastpath: Refer to The Source Of GL Accounts On Financial Transactions for a description of where the system extracts the distribution codes used to construct the GL accounts.

Determining the Statistical Quantity

The statistical quantity for each general entry line in the financial transaction is populated by adding together the statistical quantity for all bill calculation lines for the same distribution code. The bill calculation line indicates a statistical quantity if its corresponding rate component has marked the GL statistical quantity setting.

The statistical quantity for a bill calculation line is the prorated billable quantity for that line.

- If the corresponding rate component for the bill calculation line is not marked **measures peak**, then the statistical quantity corresponds to the billable quantity for the bill line. This is because billable quantities for SQs that do not represent a peak quantity are prorated when applying the rate.
- If the corresponding rate component for the bill calculation line is marked **measures peak**, then the statistical quantity may not correspond to the billable quantity for the bill calculation line. SQs that measure peak are not prorated when applying the rate. However, when determining the statistical amount to send to the general ledger, the quantity is prorated in the same manner as quantities that do not measure peak amounts.

Note: Bill Calc Line Statistical Quantity. The statistical quantity for each bill calculation line is calculated when generating the bill segment and used to populate the statistical quantity for the FT GL entry. The calculated value is not saved in the bill calc line record.

The Source Of Bill Routing Information

When a bill is completed, the system creates a bill routing for each person linked to the account who receives a copy of the bill. The bill routing record contains the information that controls how, where and to whom a bill is sent.

The following points describe how this works:

- Each person who receives a copy of an account's bill is listed on the account's person information.
- A bill routing is created for each such person when the bill is completed.
- The information that appears on the bill routing record is controlled by the **bill route type** specified on the respective account / person information:

Copyright © 2024, Oracle and/or its affiliates.

- If the bill route type indicates the person's bills are routed via *fax*, the "address line 1" address constituent is populated with the person's fax number. The system knows which of a person's phone numbers is a fax number by the phone type. If the person has multiple fax numbers, one is selected at random.
- If the bill route type indicates the person's bills are routed via *email*, the "address line 1" address constituent is populated with the person's *Email address*.
- If the bill route type indicates the person's bills are routed via the *postal* service, the address constituents are populated with the address specified on the account / person's **bill address source**. Note, if the person has a seasonal address effective on the business date, the seasonal address will be used regardless of the value of **bill address source**.

Fastpath: Refer to Printing Bills and How To Reprint A Bill (For The Original Recipients or For Someone New) for more information. Refer to Account - Person Information for information regarding how to control who receives a copy of a bill, where the bill is sent, and how the bill is formatted.

Bill Messages

The topics in this section describe how messages are linked to a bill and bill segment.

The Source Of Bill Messages

When a bill is completed, the system sweeps bill messages from the following sources onto the bill.

Note: Bill or Bill Segment Messages. Bill messages will be linked either to the bill or to one its bill segments depending on the source of the bill message code. In other words, messages associated with a contract (directly or indirectly) will be linked to the bill segment; messages associated with an account (directly or indirectly) will be linked to the bill.

- All permanent and temporary bill messages linked to an account are linked to the *bill* when it is completed. Refer to Account - Bill Messages for more information. Note well that all temporary bill messages are removed from the account when they are swept onto a bill (a temporary message is swept onto the next bill produced for the account).
- The system checks if the bill's account's customer class has bill messages. If so, it links all such messages that are effective on the bill date to the *bill* when it is completed. Refer to Setting Up Customer Class Bill Messages.
- All permanent and temporary bill messages linked to contracts that contributed bill segments to the bill are linked to their respective *bill segment* when the bill is completed. Refer to Contract - Miscellaneous for more information. Note well that all temporary bill messages are removed from the contract when they are swept onto a bill segment (a temporary message is swept onto the next bill segment produced for the contract).
- The system checks if each bill segment's rate has bill messages. If so, it links all such messages that are effective on the bill segment's start date to the *bill segment* when the bill is completed. Refer to *Rate Schedule - Bill Messages*.

In addition, a user may manually add an ad hoc message to a bill.

And finally, you can develop your own background processes and algorithms that add bill messages to accounts, contracts, bills and/or bill segments. Refer to Substituting Field Values Into A Bill Message for examples.

Substituting Field Values Into A Message

Many bill messages contain static text (i.e., the message is the same on every bill). However, the system supports messages whose contents are dynamic. For example, consider the bill message `Thank you for your charitable contributions in 2000 for $123.12, please keep this bill for tax purposes`. This message contains two substitution values (the year and the amount) as is therefore considered dynamic.

Dynamic messages can be implemented as follows:

- Create a bill message code whose Message on Bill contains substitution values. For example, the bill message code to produce the message illustrated above would contain the following Message on Bill - `Thank you for your charitable contributions in %1 for %2, please keep this bill for tax purposes`.
- Use either of the following methods to link the message code and its substitution parameters to the bill:

Copyright © 2024, Oracle and/or its affiliates.

- Create a background process or algorithm to insert a permanent or temporary bill message on the appropriate accounts. Then, when an account's bill is next completed, the system will sweep the message (and its substitution parameters onto the bill).

In addition to inserting the appropriate bill message code, your background process / algorithm must also insert the appropriate substitution values. The name of the table in which account messages are inserted is CI_ACCT_MSG . The name of the table in which a message's substitution values are inserted is CI_ACCT_MSG_PRM (you will insert one row for each substitution field).

Please see the background process identified by the process ID `REACH` for an example of a background process that performs this type of function. Please see the algorithm identified by `DEP PIF MSG` for an example of an algorithm that performs this type of function.

- Rather than adding the dynamic message to account (and then letting the bill completion logic transfer the account message to the bill), you could construct a bill completion algorithm that adds the message code and substitution values on the bill (during bill completion). The name of the table in which bill messages are inserted is CI_BILL_MSGS . The name of the table in which a message's substitution values are substituted is CI_BILL_MSG_PRM (you will insert one row for each substitution field).

A Bill May Affect More Than Just Customer Balances

The topics in this section provide information about obscure things that may happen when a bill is created, frozen, or canceled.

Auto Cancel / Rebill If A Contract Is Stopped Retroactively

If the system detects a bill segment with a bill date after the stop date of the contract (because the contract was retroactively stopped), the system automatically cancels the bill segment and creates a new one with an end date equal to the contract's stop date.

Deposits May Be Refunded

Fastpath: Refer to Refunding Deposits for how deposits may be refunded when a deposit contract is "final billed".

FT Freeze Repercussions

Fastpath: Refer to Obscure Things That Can Happen for more information about things that can happen when an FT is frozen (and FT's get frozen during billing).

Budget Billing

The topics in this section provide background information about a variety of budget billing issues.

Fastpath: We strongly recommend familiarizing yourself with the topics described in Designing and Defining Budget Plans to fully appreciate how budgets fit into the system's financial architecture.

Account versus Contract

Customers who pay on a budget plan tend to believe that there is one budget amount applied to their account's bill each month. This is not true. Rather, each contract has an individual budget amount (stored in the recurring charge array). The sum of an account's contracts' budget amount is the account's budget amount.

Why does each contract have a unique budget amount? Because all financial accounting in the system is performed at the contract level, not for an entire account.

Copyright © 2024, Oracle and/or its affiliates.

While a unique budget amount is defined for each contract, a user can use a simple page to view and change the budget amount for all eligible contracts linked to an account.

It's important to be aware that some contracts linked to an account may not be eligible for budget plans. For example, assume you have a customer with a deposit contract and one or more financial services contracts. This customer will be billed for their deposit regardless of the financial service contracts' budget amounts. You define whether a contract may participate in a budget plan on the contract's contract type.

The Financial Impact Of Budget Plans

The only difference between a customer who participates in budget billing and one who doesn't is that budget billing customer have bill segments where payoff amount differs from current amount. Why? Because the payoff amount is the actual amount of the bill. The current amount is the amount the customer is expected to pay (i.e., their budget amount).

Let's run through an example of a customer on a budget to illustrate a contract where these two balances are not the same. The values in the payoff balance and current balance columns reflect the amount due after the financial transaction has been applied:

Date	Financial Transaction	Payoff Balance	Current Balance
1-Jan-99	Bill: $125, Budget $150	125	150
15-Jan-99	Payment: $150	-25	0
2-Feb-99	Bill: $175, Budget $150	150	150
14-Feb-99	Payment: $150	0	0
3-Mar-99	Bill: $200, Budget $150	200	150
15-Mar-99	Payment: $150	50	0

Fastpath: For more information about current and payoff amounts, refer to Current Balance versus Payoff Balance.

Budget Billing Background Processes

The topics in this section describe the background processes that automate management of customer who participate in budget billing plans.

Note: Budget Processing. The budget background processes only calculate budget amounts for pending start, active and pending stop contracts whose start and end date (if any) includes the budget date. Additionally, the budget date is used to select the rate schedule that is in effect for a contract.

Budget True Up

The Budget True Up background process (referred to by the batch control ID of `BUDTRUP`) examines all accounts who participate in budget billing plans. If enough time has passed since the customer's budget was last trued up, the system will calculate the account's recommended budget amount. If the recommended amount differs from their existing budget amount by more than a given tolerance percentage, the customer's budget amount (actually, the recurring charge amount on the account's contracts) will be changed.

Fastpath: The amount of time between true ups and the true up method used to true up is defined on the account's budget plan. The tolerance percentages are defined as parameters of the true-up algorithm that is plugged in on the account's budget plan. Refer to Designing and Defining Budget Plans for all the details.

Budget Monitor

The Budget Monitor background process (referred to by the batch control ID of `BUDMON`) examines all accounts that participate in budget billing plans. If the account's existing budget amount is out-of-sync with predefined tolerance percentages, the system will highlight the account on the Budget Review page.

Copyright © 2024, Oracle and/or its affiliates.

Fastpath: The tolerance percentages are defined as parameters of the budget monitor algorithm that is plugged in on the account's budget plan. Refer to Designing and Defining Budget Plans for all the details.

Canceling A Budget

The following points describe when a budget is cancelled:

- The system may cancel a budget during a collection process (if the collection process contains a Cancel Budget collection event). Refer to Types Of Collection Events for more information.
- When a contract is stopped, the system automatically cancels the budget.

When a customer's budget is cancelled, the following events transpire:

- Any recurring charge entry whose effective date is after the cancel date is deleted. This is possible if you are entering a back-dated stop date for the contract. For example, perhaps you run the Budget True Up process, which adds a new budget amount to the contract (effective today). Then imagine that the customer calls you and informs you that they moved out last week. If you stop the customer's service as of last week, the recurring charge effective today is deleted when the stop is finalized.
- A new recurring charge amount of zero is created for each of the customer's contract.
- Each contract's current balance is synchronized with its payoff balance (using the sync adjustment code defined on the contract's contract type).

Using Billable Charges For Pass Through / Convergent Billing

The term "pass through" billing is used to describe the practice of receiving charges calculated by third parties and presenting them on the customer's bill along with your own charges. "Pass through" billing is implemented in the system using Billable Charges.

The following points provide information to help you decide the most appropriate way to implement "pass through" billing given your specific requirements:

- Customers with pass through charges will need a separate contract to hold the pass through charges. We refer to this type of contract as a "billable charge" contract.

Note: A contract's contract type controls whether a contract can have billable charges linked to it. Specifically, the contract type must have a "special role" of `Billable Charge`.

- A billable charge contract holds the billable charges until the customer is next billed. At that time, a separate bill segment will be created for each unbilled billable charge linked to the billable charge contract.
- While it is not required, we recommend creating a separate billable charge contract for each type of pass through charge.
- You can interface billable charges using the Billable Charge Interface. The interfaced charges can consist of any of the following:
  - Pre-calculated bill lines that will be presented "as is" to the customer.
  - Service quantities that are used by the system to calculate the charges on behalf of the third party.
  - All of the above. You might interface both bill lines and service quantities if the third party calculated the charges but did not calculate the taxes. The dollar amount for which taxes will be calculated would be defined using a service quantity.

Note: You cannot interface interval data via the billable charge interface. Rather, you would need to interface the interval data using the standard interval data set interface and then let the system calculate the charges for you.

- The bill lines on a billable charge can fit into any of the following categories:
  - Memo-only (i.e., don't affect the general ledger). A bill line that is memo-only contain information that is purely informational. You can use memo-only bill lines to show this information on the customer's bill. If a bill line is

Copyright © 2024, Oracle and/or its affiliates.

not memo-only (i.e., it affect the general ledger), a distribution code must also be specified to define how the lines affects the general ledger.
- Show on the customer bill. It is possible to create bill lines that affect the general ledger, but are not shown on the customer's bill. This is an unusual practice, but it happens.
- Summary / detail. Each bill line has an indication of whether it is a summary or detail line. This only impacts bill presentation.

Note: The above indicators can be defaulted onto a billable charge line by using *Billable Charge Line Types* when you interface the lines into the system.

- Characteristics (i.e., user-defined fields) can be associated with billable charge lines. You might populate this information if you are interfaced information that may be useful during bill presentation or for reporting purposes.
- If service quantities are specified on a billable charge, they are saved on the bill segment created when the billable charge is billed. This means that the system can support creation of a graph of historical usage on the bill, as well as other queries on customer usage. In addition, it might be possible to event estimate product/charge type consumption based on these service quantities (but such a plug-in is not supplied with the base package).
- Cancel / rebill for billable charges is quite different than for normal bill segments. A billable charge bill segment can be cancelled, and this will reverse the financial effects of the billable charge. But without new information from the source, there is no way to rebill the customer. Therefore, if the original charges were incorrect, the source system would send both a reversal of the charges and a newly revised set of information. These could be passed as two separate billable charges or they could combined on a single billable charge.
- For most other functionality, the billable charge contract supports the same functionality as normal contracts. This includes budget billing, payment distribution, credit & collection tracking and events, current & payoff balances, movement of old debt onto payment arrangements, etc.

Fastpath: For more information about billing billable charges, refer to How To Create A One-Time Invoice. For more information about creating a billable charge, refer to Maintaining Billable Charges. For more information about interfacing a billable charge from an external system, refer to Uploading Billable Charges.

Printing Bills

The contents of this section describe the technical implementation of both an online and batch bill production.

Bill Routings Are Created For Each Recipient Of A Bill

Refer to The Source Of Bill Routing Information for a description of how the system constructs the information used to route a bill to one or more recipients.

Bill Route Types Control The Information Merged Onto Bills

Every bill routing record references a Bill Route Type. The bill route type controls the following:

- It contains an algorithm that is responsible for extracting the information merged onto your bills. Algorithms of this type are called under the following scenarios:
  - The *background process* that interacts with your bill print software calls these algorithms.
  - If your bill print software has the ability to construct a real-time image of a bill, you can plug-in an `Online Bill Image` algorithm on the Installation Record. This type of algorithm will call the extract algorithm for an appropriate bill route type to extract the information that is merged onto the bill. Refer to Technical Implementation Of Online Bill Image for the details.
- It contains the ID of the background process responsible for interacting with your bill print software in batch.

Technical Implementation Of Online Bill Images

Users can view an image of any bill that is sent to a customer on Bill - Main if you set up the following:

Copyright © 2024, Oracle and/or its affiliates.

- Plug-in an `Online Bill Display` construction algorithm on the Installation Record.
- Plug-in the appropriate bill extract algorithm on each *Bill Route Type*.

The system provides algorithms that interact with bill print software that renders an image of the bill in a PDF. The following points describe what takes place when clicking **Display Bill** when these sample algorithms are used.

- The sample `Online Bill Display` algorithm ONLN-BL-DSP is executed.
- This algorithm calls the bill extract algorithm for an appropriate bill route type (as determined by the algorithm). The sample bill extract algorithm BLEX-EX constructs the information that appears on the bill and returns it to the `Online Bill Display` algorithm. This algorithm, in turn, passes it to your bill print software.

Fastpath: Refer to The Doc 1 Records for examples of the record layouts passed to the Doc 1 print software.

- Your bill print software renders an image of the bill in a PDF and returns it to the `Online Bill Display` algorithm.
- And finally, the system displays the PDF in a separate browser session. Note that the client must have Adobe Acrobat Reader installed to view PDF files.

The system provides algorithms that interact with BI Publisher or Crystal Reports to render an online image of your bill. The following points describe what takes place when clicking **Display Bill** when these sample algorithms are used.

- The sample `Online Bill Display` algorithm C1-ONLIN-CR is executed.
- This algorithm calls the bill extract algorithm for an appropriate bill route type (as determined by the algorithm). The sample bill extract algorithm C1-BLEX-CR is supplied for displaying bills via BI Publisher or Crystal. This is not actually extracting data because when you use BI Publisher or Crystal reports, the report itself is responsible for extracting and formatting the data. The base product algorithm supplied returns the appropriate report code to the `Online Bill Display` algorithm, which in turn returns the report code to the bill page.
- A user exit on the bill page invokes BI Publisher or Crystal and displays the bill in a separate browser session.

Technical Implementation Of Printing Bills In Batch

The batch process that extracts bill information reads all bill routing records in a given run number that are marked with its batch control ID

The base package example of this process (known by the batch control ID of *POSTROUT*) simply calls the extract algorithm on the routing record's route type to format the information placed onto the flat file. Refer to Bill Route Types Control The Information On Bills for more information.

The base package also supplies a batch bill print production (known by the batch control ID of RTTYPOST) specifically used to print bills via Crystal reports. This process invokes every distinct extract algorithm found on route types referring to this batch control id. The extract algorithms are responsible for building the information needed to schedule the report to run in Crystal and submitting the report to be scheduled.

Note: If your software doesn't support online bill images. The algorithm that formats bill extract records that's plugged in on *bill route type* serves two purposes: 1) it interacts with the online bill display algorithm plugged into to the installation record to support online images of a bill, and 2) it interacts with the background process to support printing your bills in batch. If your bill print software does not support the rendering of bill images real time, there is no need to create an extract algorithm. Rather, you should simply develop your own download process that works with your bill print software to print bills in batch (and then specify this batch process on your bill route type).

Reproducing The Bill Print Flat File

You can reproduce the flat file at any time. Simply request the download process and specify the run number associated with the historic run.

How To Reprint A Specific Bill

Refer to How To Reprint A Bill (For The Original Recipients or For Someone New) for instructions describing how to reprint a specific bill.

Copyright © 2024, Oracle and/or its affiliates.

Who Gets A Copy Of A Bill?

A copy of a bill is sent to every individual / business specified in the bill's routing details.

There are two ways in which a bill routing detail can be created:

- When a bill is completed, the system creates a routing detail for every person linked to the account that wants to receive a copy of the bill (as specified on Account - Person Information). Refer to The Source Of Bill Routing Information for more information.
- After a bill is completed, you may insert a bill routing record. Refer to How To Reprint A Bill (For The Original Recipients or For Someone New) for more information.

Final Bills and Bill Print

If a bill is produced for an account where all of the accounts contract's are `stopped`, `closed` or `cancelled`; the bill is flagged as being the final bill on the flat file produced by the postal routing process (refer to Bill Record - 0100 for the details of this switch).

If a bill is considered to be a final bill and the amount owing for the entire account is less than the final bill threshold amount on the installation record, the bill will be skipped (i.e., it won't appear on the flat file produced by the postal routing process). Note, this logic also suppresses bills from being produced when a payment is received AFTER the final bill and the account's balance falls beneath the installation record's final bill threshold amount. Refer to Installation Options - Billing for more information.

The Doc 1 Records

Numerous different types of records are interfaced to the Doc 1 bill print software. However, they all share a common structure:

- The first 4 bytes are called the **Printer Key**. This field is a record type used by the Doc 1 bill template.
- The next 184 bytes are called the **Sort Key**. This field is used to ensure the various records used to construct a printed bill appear in the correct order in the interface file.
- The next 12 bytes are called the **Mailing Key**. This field is used to control the order in which the formatted bills are printed.
- The next 3 bytes are a filler that is available for future use.
- The remaining bytes contain the **Bill Information** that appears on the printed bill. The type of information differs for each type of record.

The topics in this section describe each component.

Printer Keys (Record Types)

The following table defines each printer key (i.e., record type) on the bill print flat file.

Record Type	Description	Notes
0010	Global extract information	Required. 1 per flat file.
0100	Bill record	Required. 1 per bill.
0300	Current charge record	1 per current charge.
0600	Payment record	1 per payment.
1100	Contract record	1 per contract
1300	Bill segment record	1 per bill segment linked to the contract.
1500	Bill calculation header	1 per bill calculation header linked to the bill segment.

Copyright © 2024, Oracle and/or its affiliates.

Record Type	Description	Notes
1700	Bill calc line	1 per bill calculation line linked to the bill calculation header
1900	End of record type 1500	1 for each bill calculation header
2400	Service quantity detail	1 for each service quantity detail linked to the bill segment
2700	Item detail	1 for each item detail linked to the bill segment
2840	Bill segment messages	1 for each message detail linked to the bill segment
2900	End of record type 1300	1 for each bill segment
3100	Consumption history (for graph)	1 for each historical consumption period that appears on the graph
3400	Adjustment detail	1 for each adjustment
3600	End of record type 1100	1 for each contract
3900	Bill message detail	1 for each message
9999	End of record type 0100	1 for each bill

Sort Key

The following table defines the structure of the sort key that appears on each bill print record. Please note that different components are filled in for each printer key.

Field Name	Format	Source/Value/Description
Bill Print Intercept	A8	This is the user ID, if any, who reviews the bill (as specified on the account record)
Bill ID	A12	
Sequence Number	N2	The sequence number of the account / person information associated with this bill routing information.
Copy Number	N1	The number of copies desired by the account / person associated with this bill routing information.
Bill record group	A2	10 Bill 20 Payment 30 Current charge 40 Message 70 Bill end
Filler	A2	
State	A6	N/A
City	A30	N/A
Address 1	A64	N/A

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
Id	A10	N/A
Record group	A2	N/A
Contract Print Priority	A2	
Contract Id	A10	
Contract Group	A2	For Non-Location based records: 10 Contract 20 Current Charge 30 Adjustment 40 Contract end record
Bill Segment End Date	A10	If the record is related to an adjustment, this is the adjustment creation date. If the record is for histogram detail, this contains the histogram detail number.
Bill Segment Id	A12	If the record is related to an adjustment, this is the adjustment ID.
Bill Segment Group	A2	10 Bill segment 20 Bill segment calculation group 30 Bill segment read detail 40 Bill segment SQ detail 50 Bill segment item detail 55 Bill segment message 60 Bill segment end
Bill Calc Header Sequence	N2	For records related to read detail, SQ detail, item detail or bill message detail, this field contains the related sequence number for that record.
Bill Calc Header Group	A2	10 Bill calculation header 20 Bill calculation line 30 Bill calculation end
Line Seq	N3	Defines order of rows for similar record types
Filler	65	Only applicable for non-location based records. (This makes up for the location address related fields that are not applicable above).

Mailing Key

The following table defines the structure of the mailing key that appears on each bill print record.

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
Postal code	A12	This is the postal code on the bill routing record.

Bill Information

The topics in this section describe the information that appears on each on each bill print record.

Global Extract Information Record (0010)

This is a "header" record that contains information about the extract run as a whole.

Note: Please be aware that if you do not sort your extract file by **CIS Sort Key** after it is produced, this record will physically be the last record on the extract file.

Field Name	Format	Source/Value/Description
BATCH_CD	A8	Batch Control Code of the extract process.
BATCH_NBR	N10	Batch Number of the extract process
BATCH_RERUN_NBR	N10	Batch Rerun number of the extract process.
BILL_CNT	N10	Count of bills on this extract. Multiple copies of a bill are counted multiple as well.
EXTRACT_DTTM	A26	System time of extraction. Also updated in bill routing table.
BILL_ID_START	A12	This is only used when a bill is produced as a result of an online request (from Bill - Main). It contains the Bill Id to extract.
BILL_ID_END	A12	This is only used when a bill is produced as a result of an online request (from Bill - Main). It contains the Bill Id to extract.
THREAD_NBR	N10	This field is only used if bills are extracted in batch. It contains the thread number in which the bill was extracted.
THREAD_CNT	N10	This field is only used if bills are extracted in batch. It contains the total number of threads that were extracted.

Bill Record (0100)

Field Name	Format	Source/Value/Description
BILL_ID	A12	CI_BILL
ACCT_ID	A10	CI_BILL
BILL_DT	D	CI_BILL
DUE_DT	D	CI_BILL
COPY_NBR	N1	
NBR_BILL_COPIES	N1	CI_BILL_RTG
BILL_PRT_INTERCEPT	A8	CI_BILL_RTG
BILL_FORMAT_FLG	A2	CI_BILL_RTG
REPRINT_SW	A1	CI_BILL_RTG
CASH_ONLY_SW	A1	
CUR_BAL	S13.2	CI_BILL_SA

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
CUR_CHG_AMT	S13.2	CI_FT
CUR_CORR_AMT	S13.2	CI_FT
CUR_ADJ_AMT	S13.2	CI_FT
CUR_PAY_AMT	S13.2	CI_FT
CUR_PREV_BAL	S13.2	CI_FT
TOT_BAL	S13.2	CI_BILL_SA
TOT_CHG_AMT	S13.2	CI_FT
TOT_CORR_AMT	S13.2	CI_FT
TOT_ADJ_AMT	S13.2	CI_FT
TOT_PAY_AMT	S13.2	CI_FT
TOT_PREV_BAL	S13.2	Computed during extract.
CURRENCY_CD	A3	CI_BILL_SA
DELINQUENT_AMT	S13.2	Computed during extract.
CUST_PO_ID	A20	CI_BILL_RTG
ACCT_ENTITY_NAME	A64	CI_PER_NAME. Account Entity Name. Might include prefix/suffix from Account/ Person if available.
ENTITY_NAME	A64	CI_BILL_RTG. Mailing Entity Name.
ADDRESS_SBR	A361	CI_BILL_RTG. Refer to its components in a later section.
FINAL_BILL_SW	A1	Computed during extract. Contains a value of Y if this bill is considered to be a "final" bill.
SCHED_EXTRACT_DT	D	CI_APAY_CLR_STG. This is the date that the automatic payment will be downloaded.
ENTITY_NAME1	A64	CI_BILL_RTG. Mailing Entity Name1.
ENTITY_NAME2	A64	CI_BILL_RTG. Mailing Entity Name2.
ENTITY_NAME3	A64	CI_BILL_RTG. Mailing Entity Name3.
ACCT-CUR-SYMBOL	A4	Currency symbol for the account's currency code.

Current Charge Record (0300)

Field Name	Format	Source/Value/Description
DEBT-CLASS-CD	A4	Debt Class of Contract Type
DEBT-CLASS-DESCR	A30	Debt Class Description
CUR-CHARGE-AMT	S13.2	Derived from CI_FT (Only the current charge bill segments.)
TOT-CHARGE-AMT	S13.2	Derived from CI_FT (Only the current charge bill segments.)

Copyright © 2024, Oracle and/or its affiliates.

Payment Record (0600)

Field Name	Format	Source/Value/Description
FT TYPE	A2	Pay or Pay Cancel ('PS' or 'PX')
PAY-DT	D	Derived from CI_FT
CAN_RSN_CD	A4	Derived from CI_PAY, Only for cancellation
CAN-RSN-DESCR	A30	Derived form CI_PAY_CAN_RSN
CUR-PAY-AMT	S13.2	Derived from CI_FT
TOT-PAY-AMT	S13.2	Derived from CI_FT

Record (0900)

Field Name	Format	Source/Value/Description
NON_PREM_SW	A1	N/A
PREM_ID	A10	N/A
ADDRESS_SBR	A361	N/A

Contract Record (1100)

Field Name	Format	Source/Value/Description
SA_ID	A10	CI_SA
OLD_ACCT_ID	A15	CI_SA
CUST_READ_FLG	A2	CI_SA
SIC_CD	A4	CI_SA
SIC_DESCR	A30	CI_SIC
BUSINESS_UNIT	A10	CI_SA
SA_TYPE_CD	A8	CI_SA
SA_TYPE_DESCR	A30	CI_SA_TYPE
SA_TYPE_DFLT_DESCR	A80	CI_SA_TYPE
SVC_TYPE_CD	A2	CI_SA_TYPE
SVC_TYPE_DESCR	A30	CI_SVC_TYPE
REV_CL_CD	A8	CI_SA_TYPE
CHAR_PREM_ID	A10	CI_SA
SA_REL_ID	A10	CI_SA_REL
SA_REL_TYPE_CD	A8	CI_SA_REL
SA_REL_TYPE_DESCR	A30	CI_SA_REL_TYPE
SPR_CD	A12	CI_SA_REL
SPR_DESCR	A50	CI_SPR
SPR_ENTITY_NAME	A64	CI_PER_NAME (this is the service provider's person's name)

Copyright © 2024, Oracle and/or its affiliates.

Bill Segment Record (1300)

Field Name	Format	Source/Value/Description
START_DT	D	CI_BSEG
END_DT	D	CI_BSEG
CLOSING_BSEG_SW	A1	CI_BSEG
EST_SW	A1	CI_BSEG
BILL_CYC_CD	A4	CI_BSEG
SQ_OVERRIDE_SW	A1	CI_BSEG
ITEM_OVERRIDE_SW	A1	CI_BSEG
BSEG_STAT_FLG	A2	CI_BSEG
CAN_RSN_CD	A4	CI_BSEG
CAN_RSN_DESCR	A30	CI_BILL_CAN_RSN
CUR_AMT	S13.2	CI_FT
TOT_AMT	S13.2	CI_FT

Bill Calc Header Record (1500)

Field Name	Format	Source/Value/Description
START_DT	D	CI_BSEG_CALC
END_DT	D	CI_BSEG_CALC
SETID	A10	CI_BSEG_CALC
RS_CD	A8	CI_BSEG_CALC
CALC_AMT	S13.2	CI_BSEG_CALC
DESCR_ON_BILL	A80	CI_BSEG_CALC

Bill Calc Line Record (1700)

Field Name	Format	Source/Value/Description
APP_IN_SUMM_SW	A1	CI_BSEG_CALC_LN
CALC_AMT	S13.2	CI_BSEG_CALC_LN
EXEMPT_AMT	S13.2	CI_BSEG_CALC_LN
UOM_CD	A4	CI_BSEG_CALC_LN
TOU_CD	A8	CI_BSEG_CALC_LN
SQI_CD	A8	CI_BSEG_CALC_LN
BILL_SQ	S12.6	CI_BSEG_CALC_LN
DESCR_ON_BILL	A80	CI_BSEG_CALC_LN

Copyright © 2024, Oracle and/or its affiliates.

End Bill Calc Header Record (1900)

Field Name	Format	Source/Value/Description
Dummy field		

Bill Segment Read Record (2100)

Field Name	Format	Source/Value/Description
SP_ID	A10	CI_BSEG_READ
BADGE_NBR	A16	CI_MTR
UOM_CD	A4	CI_BSEG_READ
TOU_CD	A8	CI_BSEG_READ
RDG_DEC_POS	N1	CI_REG. Decimal positions of the reads.
START_REG_READING	S9.6	CI_BSEG_READ
START_READ_DTTM	T	CI_BSEG_READ
START_ESTIMATE_SW	A1	CI_REG_READ
END_REG_READING	S9.6	CI_BSEG_READ
END_READ_DTTM	T	CI_BSEG_READ
END_ESTIMATE_SW	A1	CI_REG_READ
MSR_QTY	S12.6	CI_BSEG_READ
USAGE_FLG	A2	CI_BSEG_READ
USE_PCT	N3	CI_BSEG_READ
HOW_TO_USE_FLG	A2	CI_BSEG_READ
CONSUM_SUB_FLG	A2	CI_REG
MSR_PEAK_QTY_SW	A1	CI_BSEG_READ
REGISTER_CONST	S6.6	CI_BSEG_READ
QTY_DEC_POS	N1	CI_UOM. Decimal positions of the final quantity.
FINAL_UOM	A4	CI_BSEG_READ
FINAL_TOU	A8	CI_BSEG_READ
FINAL_REG_QTY	S12.6	CI_BSEG_READ
SQI_CD	A8	CI_BSEG_READ
FINAL_SQI_CD	A8	CI_BSEG_READ

Bill Segment SQ Record (2400)

Field Name	Format	Source/Value/Description
UOM_CD	A4	CI_BSEG_SQ
TOU_CD	A8	CI_BSEG_SQ

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
SQI_CD	A8	CI_BSEG_SQ
SQ_DEC_POS	N1	CI_BSEG_SQ
BILL_SQ	S12.6	CI_BSEG_SQ

Bill Segment Item Record (2700)

Field Name	Format	Source/Value/Description
START_DT	D	CI_BSEG_ITEM
END_DT	D	CI_BSEG_ITEM
ITEM_CNT	S9.2	CI_BSEG_ITEM
ITEM_DESCR	A30	CI_ITEM_TYPE

Bill Segment Message Record (2840)

Field Name	Format	Source/Value/Description
BILL_MSG_CD	A4	CI_BSEG_MSG
MSG_ON_BILL	A254	CI_BILL_MSG_L

End Bill Segment Record (2900)

Field Name	Format	Source/Value/Description
Dummy field		

Histogram Record (3100)

Field Name	Format	Source/Value/Description
UOM_CD	A4	CI_SA_TYPE
TOU_CD	A8	
SQI_CD	A8	
SQ_DEC_POS	N1	CI_UOM Decimal positions of SQ.
SQ_CNT	N2	Count of SQ (maximum of 13.)
MAX_BILL_SQ	S12.6	Maximum Bill SQ from list. This is derived from CI_BSEG_READ.
SQ_DT (1:13)	D	Date (for label of the SQ). This is derived from CI_BSEG.
BILL_SQ (1:13)	S12.6 (1:13)	13 occurrences of QTY, one for each period ending on the SQ_DT. This is derived from CI_BSEG_READ.

Adjustment Record (3400)

Field Name	Format	Source/Value/Description
ADJ_DT	D	CI_FT
FT_TYPE_FLG	A2	CI_FT

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
CUR_AMT	S13.2	CI_FT
TOT_AMT	S13.2	CI_FT
ADJ_TYPE_CD	A8	CI_ADJ
ADJ_TYPE_DESCR	A30	CI_ADJ_TYPE
DESCR_ON_BILL	A80	CI_ADJ_TYPE
CAN_RSN_CD	A4	CI_ADJ
CAN_RSN_DESCR	A30	CI_ADJ_CAN_RSN

End Contract Record (3600)

Field Name	Format	Source/Value/Description
Dummy field		

End Record (3700)

Field Name	Format	Source/Value/Description
Dummy field		

Message Record (3900)

Field Name	Format	Source/Value/Description
BILL_MSG_CD	A4	CI_BILL_MSGS
MSG_PRIORITY_FLG	A2	CI_BILL_MSG
INSERT_CD	A8	CI_BILL_MSG
MSG_ON_BILL	A254	CI_BILL_MSG

End Bill Record (9999)

Field Name	Format	Source/Value/Description
Dummy field		

Address Sub-record

The address sub-record that appears in both the bill record (0100) and the record (0900) is composed of the following:

Field Name	Format	Description
COUNTRY	A3	
ADDRESS1	A64	Refer to The Source Of Bill Routing Information for a description of conditions when this field will contain a fax number or an Email address rather than the first line of a postal address (this only occurs for a bill record (0100)).
ADDRESS2	A64	

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Format	Description
ADDRESS3	A64	
ADDRESS4	A64	
CITY	A30	
NUM1	A6	
NUM2	A4	
HOUSE_TYPE	A2	
COUNTY	A30	
STATE	A6	
POSTAL	A12	
GEO_CODE	A11	
IN_CITY_LIMIT	A1	

Writing Off Bills

If your organization practices open-item accounting and your credit and collections rules revolve around collecting on unpaid bills, you can set up the system to automatically write-off bills.

If you've enabled this functionality, Bill - Main will contain a button that a user can click to manually write off a bill. A corresponding button exists that a user can click to reverse a written-off bill.

Fastpath: Refer to Writing Off Bills for a description of how to set up the system to allow both automated and manual bill write off.

Idiosyncratic Manual Bill Cancellation

If a bill's account is associated with a Customer Class that has a `Cancel Bill` algorithm, the system invokes this algorithm before the bill is displayed to determine if the bill is "cancellable". If the algorithm indicates the bill is cancellable, a button appears on Bill - Main. When clicked, the `Cancel Bill` algorithm is invoked again to cancel the bill. This functionality is meant to support unique cancellation needs required by some implementations; the sample base-package algorithm is empty.

Suppressing Extraction for Zero or Negligible Amount Bills

Oracle Revenue Management and Billing enables you to suppress the extraction process for bills when the bill amount is zero or within the tolerance limit. This feature helps to reduce the cost spent on sending zero or negligible amount bills to customers. To enable this feature for an account, you need to create an algorithm using the **C1-SUPPBILLS** algorithm type and attach it to the **Post-Bill Completion** system event of the respective customer class. This algorithm contains the following parameters:

- **Consider Threshold Limit** - Used to indicate whether you want to suppress the extraction of bills when the bill amount is within the threshold limit. The valid values are:
  - **Y** - Used to indicate that you want to suppress the extraction of bills when the bill amount is within the threshold limit.
  - **N** - Used to indicate that you want to suppress the extraction of only those bills where the bill amount is zero.

Copyright © 2024, Oracle and/or its affiliates.

This parameter is mandatory.

- **Routing Threshold Debit Amount** - Used to specify the debit threshold amount. The value can be zero or a positive integer. This parameter is required when the **Consider Threshold Limit** parameter is set to **Y**.
- **Routing Threshold Credit Amount** - Used to specify the credit threshold amount. The value can be a negative integer. This parameter is required when the **Consider Threshold Limit** parameter is set to **Y**.
- **Consider Ledger Impact** - Used to indicate whether you want to suppress the extraction of bills when the GL entry does not exist for the bill. The valid values are:
  - **Y** - Used to indicate that you do not want to suppress the extraction of bills when the GL entry exists for the bill.
  - **N** - Used to indicate that you want to suppress the extraction of bills irrespective whether the GL entry exists or not for the bill.

  This parameter is mandatory.
- **Routing Type To Be Suppressed** - Used to indicate the bill route type for which you want to suppress the extraction of bills. You can specify multiple bill route types separated by a comma. If you do not specify the value for this parameter, the system will suppress the extraction of bills for all bill route types when this algorithm is attached on the customer class.

Note: The system assumes that the debit and credit threshold amount are specified in the bill currency.

Based on the above parameters defined in the algorithm, the system decides whether the bill must be suppressed for the bill routing. If a bill must be suppressed for the extraction process, the algorithm will select the **Do Not Extract** option for the required bill route type on the bill completion. Let us understand this with the help of the following examples:

- Example 1
- Example 2
- Example 3

In addition, the system will set the following columns corresponding to the bill routing information of the bill in the **CI_BILL_ROUTING** table:

Column	Column Value
BATCH_CD	' '
BATCH_NBR	0
NO_BATCH_PRT_SW	'Y'
BILL_RTE_TYPE_CD	Here, the column value is set to the bill route types (separated by comma) for which the extraction of bills is suppressed. **Note:** If the extraction of bills is suppressed for all bill route types, the column value would be NULL.

If required, you can deselect the **Do No Extract** option for a bill route type whenever required for a bill from the **Bill** screen.

Example 1

If you set the following values in the **C1-SUPPBILLS** algorithm:

- **Consider Threshold Limit** - Y
- **Routing Threshold Debit Amount** - 5
- **Routing Threshold Credit Amount** - -5
- **Consider Ledger Impact** - Y
- **Routing Type To Be Suppressed** - Postal

Then, the system will behave in the following manner:

Copyright © 2024, Oracle and/or its affiliates.

Bill	Bill Amount	GL Exists	Select the Do No Extract Option
BILL 1	0	No	Yes
BILL 2	0	Yes	No
BILL 3	3	No	Yes
BILL 4	3	Yes	No
BILL 5	-4	No	Yes
BILL 6	-4	Yes	No
BILL 7	10	No	No
BILL 8	10	Yes	No
BILL 9	-16	No	No
BILL 10	-16	Yes	No

As mentioned in the above table, the system will accordingly select the **Do No Extract** option for the **Postal** bill route type.

Example 2

If you set the following values in the **C1-SUPPBILLS** algorithm:

- **Consider Threshold Limit** - Y
- **Routing Threshold Debit Amount** - 5
- **Routing Threshold Credit Amount** - -5
- **Consider Ledger Impact** - N
- **Routing Type To Be Suppressed** - Postal, Email

Then, the system will behave in the following manner:

Bill	Bill Amount	GL Exists	Select the Do No Extract Option
BILL 1	0	No	Yes
BILL 2	0	Yes	Yes
BILL 3	3	No	Yes
BILL 4	3	Yes	Yes
BILL 5	-4	No	Yes
BILL 6	-4	Yes	Yes
BILL 7	10	No	No
BILL 8	10	Yes	No
BILL 9	-16	No	No
BILL 10	-16	Yes	No

As mentioned in the above table, the system will accordingly select the **Do No Extract** option for the **Postal** and **Email** bill route types.

Example 3

Copyright © 2024, Oracle and/or its affiliates.

If you set the following values in the **C1-SUPPBILLS** algorithm:

- **Consider Threshold Limit** - N
- **Consider Ledger Impact** - Y

Then, the system will behave in the following manner:

Bill	Bill Amount	GL Exists	Select the Do No Extract Option
BILL 1	0	No	Yes
BILL 2	0	Yes	No
BILL 3	3	No	No
BILL 4	3	Yes	No
BILL 5	-4	No	No
BILL 6	-4	Yes	No

As mentioned in the above table, the system will accordingly select the **Do No Extract** option for all bill route types.

Due Date based on Bill Creation or Completion Date

Oracle Revenue Management and Billing enables you to derive the bill due date for an account using the bill creation or completion date. In order to calculate the bill due date for an account using the bill creation or completion date, you need to attach an algorithm which is created using the **C1DUDTCALC** algorithm type to the **Override Due Date** system event of the respective customer class. This algorithm contains the following parameters:

- **Days To Add** - Used to specify the characteristic type which indicates the number of days you want to add to the bill creation date or bill completion date in order to derive the bill due date. This type of characteristic is defined for an account.
- **Specific Date** - Used when you want a specific date to be a bill due date. In such case, the month and year for the bill due date are derived from the bill creation date or bill completion date. This parameter is used when the characteristic type specified in the **Days To Add** parameter is not defined for the account. If the derived bill due date is earlier than bill creation date or bill completion date, the system adds one month to the bill due date. Here, the parameter value should be greater than zero and less than or equal to 28.
- **Due Date Rule** - Used to specify the characteristic type which indicates whether the bill due date should be calculated considering the business days (i.e. excluding weekends and holidays) or considering the calendar days (i.e. including weekends and holidays). This type of characteristic is defined for a division. The system derives the characteristic from the division to which the account belongs. Here, you must specify a predefined characteristic type which has the following two characteristic values:

Characteristic Value	Should be…
W	Used when you want to calculate the bill due date considering the business days (i.e. excluding weekends and holidays). The system derives the business days from the work calendar of the division. If the holidays are defined in the work calendar of the division, the system excludes the holidays along with weekends during bill due date calculation.
C	Used when you want to calculate the bill due date considering the calendar days (i.e. including weekends and holidays). The system derives the calendar days from the work calendar of the division.

- **Date To Consider** - Used to indicate whether you want to calculate the bill due date using the bill creation date or bill completion date. The valid values are:
  - **BILL_CRE_DT** - Used when you want to calculate the bill due date using the bill creation date.

Copyright © 2024, Oracle and/or its affiliates.

- **BILL_COMP_DT** - Used when you want to calculate the bill due date using the bill completion date.

Adjust Underpayments at Bill Level through Write Off

Until now, you were able to write off at the account level where the system automatically created underpayment or short payment adjustments when the outstanding balance of the account was within the threshold limit. This helped to handle the underpayment or short payment scenarios at the account level. Now, in addition, the similar feature is offered at the bill level. Now, you can write off at the bill level where the system automatically creates underpayment or short payment adjustments when the unpaid balance of the bill is within the threshold limit. Note that the short payment adjustments to write off the unpaid bill amount can be created against the payment or payment segment. There can be a rare scenario where a bill segment does not have any pay segment but is still eligible for write off. In such cases, the system will directly create short payment adjustments to write off the unpaid bill amount against the payment.

The **C1-PDOV-BSDT** algorithm type is enhanced to create underpayment or short payment adjustments when the unpaid balance of the bill is within the threshold limit. If you want to create underpayment or short payment adjustments at the bill level, you need to do the following:

- Distribute payments using a match type where the **C1-PDOV-BSDT** algorithm is attached to the **Pay Dist Override Algorithm** spot.
- Set the value for all the newly introduced parameters in the **C1-PDOV-BSDT** algorithm:
  - **Underpayment Tolerance Type - F (Fixed) or P (Percentage)** - Used to determine whether the specified tolerance is a fixed amount or a percentage value. The valid values are:
    - F
    - P
  - **Underpayment Tolerance Fixed Amount/Percentage** - Used to specify the unpaid flat amount or unpaid percentage value (i.e. unpaid amount with respect to original amount). The system then automatically creates underpayment or short payment adjustments when the unpaid balance of the bill is within the threshold limit.
  - **Underpayment Adjustment Type** - Used to specify the adjustment type using which you want to create the underpayment or short payment adjustments to write off the unpaid amount of a bill.

  All these parameters are optional. If you specify a value for any of the above three parameters, you need to specify the value for all the above three parameters. If you do not set the above three parameters in the **C1-PDOV-BSDT** algorithm, the system will not create underpayment or short payment adjustments at the bill level.

If you set the above two pre-requisites, the system will create the underpayment or short payment adjustments to write off the unpaid amount of a bill not only from the user interface but also through the **C1-PaymentsService** or **C1-PaymentService** inbound web service.

Note that while calculating the underpayment threshold amount, the system derives the unpaid debit amount after offsetting the debit amount with the credit amount on the bill. The system then determines the underpayment threshold amount using the unpaid debit amount and underpayment tolerance amount or percentage. The way system derives the underpayment threshold amount differs depending on two factors - whether the underpayment tolerance is flat or percentage value and whether the underpayment tolerance flat amount is greater than, equal to, or less than the unpaid debit amount. If the underpayment tolerance flat amount is greater than or equal to the unpaid debit amount, the system sets the underpayment threshold amount to the unpaid debit amount. However, if the underpayment tolerance flat amount is less than the unpaid debit amount, the system sets the underpayment threshold amount to unpaid debit amount minus underpayment tolerance flat amount. If the payment amount is greater than or equal to the underpayment threshold amount and if the payment amount is less than the unpaid debit amount, the system will create underpayment or short payment adjustment at the bill level.

Let us understand this with the help of the following scenarios:

Scenario 1			**Comments**
Flat Underpayment Tolerance Scenario 1	Original Bill Amount	100	If the underpayment tolerance amount (i.e. 10) is less than the debit amount on the bill (i.e. 80),
	Credit Amount on the Bill	-20	

Copyright © 2024, Oracle and/or its affiliates.

Scenario 1			**Comments**
	Debit Amount on the Bill	80 (i.e. 100-20)	then the underpayment threshold amount is debit amount on the bill minus underpayment tolerance amount (i.e. 80-10=70). Here, the system will create underpayment or short payment adjustment at the bill level because the payment amount is greater than or equal to the underpayment threshold amount and the payment amount is less than the debit amount on the bill.
	Payment Amount	75	
	Unpaid Bill Amount	5 (i.e. 80-75)	
	Underpayment Tolerance Amount	10	
	Underpayment Threshold Amount	80-10=70	
Flat Underpayment Tolerance Scenario 2	Original Bill Amount	150	If the underpayment tolerance amount (i.e. 150) is greater than or equal to the debit amount on the bill (i.e. 150), then the underpayment threshold amount is equal to debit amount on the bill (i.e. 150). Here, the system will not create underpayment or short payment adjustment at the bill level because the payment amount is less than the underpayment threshold amount.
	Credit Amount on the Bill	0	
	Debit Amount on the Bill	150 (i.e. 150-0)	
	Payment Amount	0	
	Unpaid Bill Amount	150 (i.e. 150-0)	
	Underpayment Tolerance Amount	150	
	Underpayment Threshold Amount	150	
Percentage Underpayment Tolerance Scenario 3	Original Bill Amount	100	Underpayment threshold amount = ((100 - Underpayment Tolerance Percentage)/100 * Debit Amount on the Bill). Here, the system will create underpayment or short payment adjustment at the bill level because the payment amount is greater than or equal to the underpayment threshold amount and the payment amount is less than the debit amount on the bill.
	Credit Amount on the Bill	-20	
	Debit Amount on the Bill	80 (i.e. 100-20)	
	Payment Amount	75	
	Unpaid Bill Amount	5 (i.e. 80-75)	
	Underpayment Tolerance Percentage	50	
	Underpayment Threshold Amount	(100 - 50)/100) * 80 = 40	

Default Bill Cutoff Date

Until now, on clicking the **Generate** button in the **Bill** screen, the **Generate** window appeared where the cutoff date was set to the system date. Now, to address some of the business requirements, the system will drive the default cutoff date based on the below stated logic.

If one or more completed bills exist for the account, the system will consider all its frozen bill segments and derive the start or end date of the bill segment with the latest date. The system considers the bill segment start or end date depending on whether the **Billable Charges Date - Non Recurring** option type of the **C1_FINTRANOP** feature configuration is set to **S** or **E**, respectively.

If the **Billable Charges Date - Non Recurring** option type of the **C1_FINTRANOP** feature configuration is set to **S**, the system will derive the start date of the frozen bill segment with the latest date. Similarly, if the **Billable Charges Date - Non Recurring** option type of the **C1_FINTRANOP** feature configuration is set to **E**, the system will derive the end date of the frozen bill segment with the latest date. Once the latest frozen bill segment date of the account is available, the system derives the next window end date of the bill cycle after the latest frozen bill segment date and sets

Copyright © 2024, Oracle and/or its affiliates.

it as the default cutoff date. However, if the bill cycle is not defined for the account or if the next window is not defined in the bill cycle, the system sets the system date as the default cutoff date.

If you are generating a bill for an account for the first time or if all the bills of the account are canceled for any reason, the system considers the account setup date or bill on or after date whichever is latest. If the bill on or after date is not defined for the account or if the latest date is the account setup date, the system derives the next window end date of the bill cycle after the account setup date and sets it as the default cutoff date. If the latest date is the bill on or after date and if the next window end date after the latest date is earlier than the system date, the system sets the next window end date as the default cutoff date. However, if the latest date is the bill on or after date and if the next window end date after the latest date is later than the system date, the system sets the system date as the default cutoff date.

Let us understand the above stated logic with the help of the following scenarios. Here, we have assumed that the **Billable Charges Date - Non Recurring** option type of the **C1_FINTRANOP** feature configuration is set to **E** in the system.

Scenario	Account	Bill	Bill Segment	Bill Segment Start Date	Bill Segment End Date	Account Setup Date	Bill On or After Date	Latest Bill Date	System Date	Derived Window	Default Cutoff Date
S1	A1	B1	BS1	01-Jan-2020	15-Jan-2020	31-Dec-2019	-	31-Mar-2020	01-May-2020	01-Apr-2020 - 30-Apr-2020	30-Apr-2020
			BS2	01-Jan-2020	01-Jan-2020						
		B2	BS3	01-Feb-2020	15-Feb-2020						
			BS4	01-Feb-2020	29-Feb-2020						
		B3	BS5	01-Mar-2020	15-Mar-2020						
			BS6	15-Mar-2020	31-Mar-2020						
S2	A2	B1	BS1	01-Jan-2020	15-Jan-2020	31-Dec-2019	-	31-Mar-2020	01-May-2020	No window is available after 31-Mar-2020 in the account's bill cycle	01-May-2020
			BS2	01-Jan-2020	01-Jan-2020						
		B2	BS3	01-Feb-2020	15-Feb-2020						
			BS4	01-Feb-2020	29-Feb-2020						
		B3	BS5	01-Mar-2020	15-Mar-2020						
			BS6	15-Mar-2020	31-Mar-2020						
S3	A3	-	-	-	-	31-Dec-2019	-	31-Dec-2019	01-May-2020	01-Jan-2020 -31-Jan-2020	31-Jan-2020
S4	A4	-	-	-	-	31-Dec-2019	31-May-2020	31-May-2020	01-Jun-2020	01-Jun-2020	01-Jun-2020

Copyright © 2024, Oracle and/or its affiliates.

Scenario	Account	Bill	Bill Segment	Bill Segment Start Date	Bill Segment End Date	Account Setup Date	Bill On or After Date	Latest Bill Date	System Date	Derived Window	Default Cutoff Date
										-30-Jun-2020	
S5	A5	-	-	-	-	31-Dec-2019	31-Mar-2020	31-Mar-2020	01-Jun-2020	01-Apr-2020 -30-Apr-2020	30-Apr-2020
S6	A6	-	-	-	-	31-Dec-2019	31-May-2020	31-May-2020	01-May-2020	-	-

Note:

Scenario 1 - The system sets the next window end date of the bill cycle after the latest frozen bill segment date as the default cutoff date.

Scenario 2 - The system sets the system date as the default cutoff date because next window does not exist in the bill cycle.

Scenario 3 - The system sets the next window end date of the bill cycle after the account setup date as the default cutoff date.

Scenario 4 - The system sets the system date as the default cutoff date because the next window end date is later than the system date.

Scenario 5 - The system sets the next window end date as the default cutoff date because it is earlier than the system date.

Scenario 6 - No default cutoff date is derived because the bill after date is later than the system date.

If the bill on or after date does not exist for the account or if the bill on or after date is earlier than or equal to than the system date, the system, by default, displays the cutoff date. An appropriate message appears in the **Generate** window that indicates how the default cutoff date is derived for the account. You can change the cutoff date, if required.

However, if the bill on or after date is later than the system date, the system, by default, does not display the cutoff date and the **Calculate** button is also disabled in the **Generate** window. Note that the **Calculate** button is disabled because you cannot generate a bill for the account until the bill on or after date. An appropriate message appears in the **Generate** window indicating the reason why the system could not derive the default cutoff date.

Note that the above default cutoff date derivation logic is only available when you create a bill for an account from the user interface.

Maintaining Bills

A bill is used to communicate changes in a customer's financial obligations to the customer. The topics in this section describe how to maintain bills.

Note: The system creates most bills behind-the-scenes. Most bills are created by the system when it processes the bill cycle schedule. You should only have to access the bill pages to correct a bill or to add a bill real-time. For information about how the system creates bills, refer to The Big Picture of Billing.

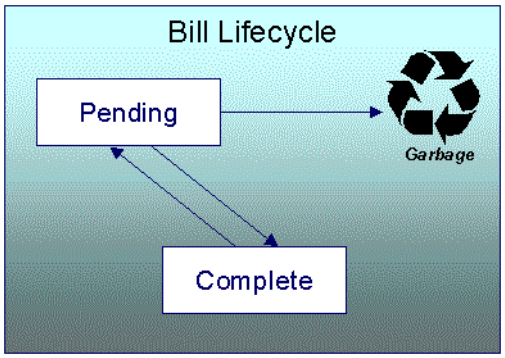
It's important to be aware that there are very few fields that are directly modifiable by a user. To modify most fields on a bill, you have to change source information (e.g., contract, rate) and then regenerate the bill. For example, if you want to change a bill's amount, you must cancel or add bill segments; you cannot change the bill's amount by modifying the

Copyright © 2024, Oracle and/or its affiliates.

bill amount field. Refer to How To for step-by-step instructions that explain how to perform common bill maintenance functions.

Bill Lifecycle

The following diagram shows the possible lifecycle of a bill.



CAUTION: This explanation only makes sense in the context of the page used to maintain bills. Refer to Bill - Main Information for the details.

A bill is initially saved in the `Pending` state. You may create one or more bill segments for the account's contracts at this point. Refer to How To for information about generating bill segments for the bill.

A bill becomes `Complete` when it is ready to be sent to the customer. Completing a bill triggers many things to occur. Refer to the section below for information about what happens when a bill is completed.

When you `complete` a bill, several things may happen:

- Pre-Bill Completion Review algorithm defined on the account's customer class is executed. If you have attached an algorithm of the C1-BILL_TOLR algorithm type on the **Pre-Bill Completion Review** system event of the account's customer class, the bill tolerance feature is enabled. For more information about the bill tolerance feature, refer to Bill Tolerance.
- Pre-completion algorithms associated with the account's contracts' contract types are executed. The system executes these algorithms first in the order of the billing processing sequence on each contract's contract type then in the order of the algorithm's sequence. These are not executed for the completion of credit notes.
- Pre-completion algorithms associated with the account's customer class are executed.
- The bill's due date is calculated. This is equal to the bill date plus the number of days defined on the account's customer class. If the resultant date is not a workday, the due date is set to the next workday. Note, this due date can be overridden if an override algorithm exists on the account's customer class.
- The bill's routing information is set up using Account - Person Information.
- If the `Freeze At Bill Completion` installation option has been turned on, `freezable` bill segments and adjustments linked to the account are frozen. Note, only `freezable` adjustments whose adjustment type indicates `Freeze At Bill Completion` will be frozen at this time. Refer to Preventing Contract Balances And The GL From Being Impacted Until Bill Completion for more information.
- If a final bill segment of a deposit contract is detected, the deposit is refunded to the customer using the refund method algorithm plugged-in on the deposit contract's deposit class. This step is skipped during the completion of credit notes.
- Post completion algorithms associated with the account's contracts' *contract types* are executed. The system executes these algorithms first in the order of the billing processing sequence on each contract's contract type then in the order of the algorithm's sequence.
- Bill completion algorithms associated with the account's customer class are executed.
- Bill messages are amalgamated from various sources and linked to the bill. Refer to The Source Of Bill Messages for more information.

Copyright © 2024, Oracle and/or its affiliates.

- Other financial transactions that have been frozen since the last bill and are marked to show on bill are linked to the new bill
  - For open-item customers adjustments and corrections are linked
  - For correction notes or credit notes, the bill cancellation financial transactions are linked
  - For all other situations, payments, adjustments and corrections are linked
- All financial transactions that don't already have a user-defined aging date will be dated with the current date. In other words, they start aging from the date the bill is completed.
- If the account pays automatically and the installation's Autopay Creation Option is to `Create At Bill Completion`, an automatic payment is created. Refer to APAYCREATE for more information about the auto pay creation algorithm. This step is skipped during the completion of credit notes.
- If the account's customer class indicates the account is an open-item customer, a match event is created. The bill's FTs and the automatic payment's FTs are linked to it. Refer to Payments and Match Events for more information about match events. This step is skipped during the completion of credit notes.
- If the account pays automatically and the installation's Autopay Creation Option is to `Create On Extract Date`, the bill is stamped with the date when the automatic payment is to be created. Refer to The System Background Processes for more information about `APAYCRET`, the background process that creates the automatic payment and `APAYDSFR`, the background process that distributes and freezes the automatic payment. This step is skipped during the completion of credit notes.
- The bill's late payment date is calculated. This is equal to the due date plus the grace days defined on the account's customer class. If the resultant date is not a workday, the LPC date is set to the next workday. This step is skipped during the completion of credit notes.
- The account is marked for review by the account debt monitor on its due date plus a grace period defined on its customer class. This step is skipped during the completion of credit notes.
- If the account's customer class indicates the account is an open-item customer, the system will create a match event if the new charges are offset by other financial transactions. Refer to Payments and Match Events for more information about match events. This step is skipped during the completion of credit notes.
- The bill's status becomes `Complete`.
- Post bill completion algorithms associated with the account's customer class are executed.

If the system cannot complete the bill (for whatever reason), the bill remains `Pending` and an error message is shown on the main bill page. After correcting the cause of the problem, attempt to complete the bill again.

A `complete` bill may be changed back to `pending` using the Reopen button on Bill - Main Information. You would reopen a bill when

- Add more bill segments to a completed bill. Refer to the How To section for information about linking bill segments for a bill.
- Fine-tune the payments, adjustments, and corrections that were linked to a completed bill. Refer to the How To section for detailed instructions.

When you're happy with the bill, you can complete it again.

Note: Cannot reopen historical bills. You may only reopen an account's most recent bill because recompleting the bill causes the ending-balance to change, and we don't want this to happen to historical bills.

Note: Automatic payments. If an automatic payment was created when the bill was completed and it has already been interfaced to the financial institution, you cannot reopen the bill. If the automatic payment exists, but it has not yet been interfaced, the system will automatically cancel the payment when you reopen the bill.

You may delete a `Pending` bill from the database. You may not delete a pending bill if: a) there are frozen bill segments linked to the bill, or b) if financial transactions were linked to the bill (and this can only happen if the bill had been previously completed).

In addition to removing the bill, the system will also remove unfrozen bill segments linked to the bill.

Copyright © 2024, Oracle and/or its affiliates.

Bill - Main Information

Open **Billing Management, Bill, Main** to maintain core bill information.

The following discussion describes the fields on this page. Refer to How To for a description of how to perform common bill maintenance functions.

Description of Page

Bill Info contains a concatenation of important information about the bill (e.g., the bill date, its status, due date, ending balance, etc.). If sequential bill numbering has been enabled, the bill's number also appears. If this bill is a correction note or credit note, a message will appear indicating this along with the ID of the original bill. If the batch billing background process generates the bill, the bill information includes the bill cycle and window start date. Bills generated manually, i.e., off-schedule, do not have this information.

Bill ID is the system-assigned unique identifier of the bill.

Account ID identifies the customer who is responsible for the bill. The name of the main customer on the account appears adjacent.

If this bill is a credit note or correction note, a message appears adjacent to the account information indicating that this bill is a credit note or correction note for another bill; the adjacent go to button will drill you to the original bill. If a credit note or correction note has been issued for one or more of this bill's bill segments, a message appears indicating such.

Bill Status is the bill's status. Refer to Bill Lifecycle for the potential values and how to handle a bill when it exists in a given state. Adjacent to the bill status is the **Display Bill** button, which will display an online image of the bill when clicked. Refer to How To Display A Bill On-line for more information.

Note: You can only use the **Display Bill** button if your system has been configured to display an on-line image; otherwise, a message indicating that the service is not available will appear. This option can only be configured by your technical staff. The system comes with configurations to display bill images using BI Publisher, DOC1, and Crystal, but can work with other applications.

Due Date is the date on which the bill is due. A bill's due date is calculated as follows:

- Add the due days specified on the account's customer class to the bill date.
- The resulting date is the due date unless it's not a workday. In this case, the due date is set to the next workday (workdays are defined in the installation options tables).
- If the account's customer class has an override due date calculation algorithm, the due date may be overridden. Refer to Setting Up Customer Classes for more information.

Note: Late payment charges. Late payment charges are calculated on the due date plus the grace period. Refer to Defining Late Payment Charge Options for more information.

Bill Date is the business date that was used when the bill was completed. For example, if batch billing starts on 11:55 pm on 1-Jan-2000 and a bill is produced at 1:00 am on 2-Jan-2000, this bill's bill date would be 1-Jan-2000.

Create Date/Time is the date and time on which the bill was originally created.

Completion Date/Time is the date and time on which the bill was completed.

If the customer pays automatically and the automatic payment has not yet been created, a brief message appears highlighting the date and amount of the future automatic payment. In addition, a button appears called **Stop Autopay**, that, when clicked, causes the associated automatic payment to stop. You might click this button if the customer indicates that they will send in a payment. After you click the **Stop Autopay** button, a message appears highlighting the stoppage. Refer to *How And When Are Automatic Payments Created* for more information about how to setup the system to defer the creation of automatic payments until the future automatic payment extract date.

In the **Message** area, a brief error message appears if there's a problem with the bill. The message area is suppressed if there are no problems with the bill.

Copyright © 2024, Oracle and/or its affiliates.

Click the magnifying button to view the long explanation. The long explanation will provide information about the cause of the error (and how to fix it).

The **Late Payment Charge Details** section contains information about how the system determines when and whether to levy late payment charges:

- **Eligible for Late Payment Charge** can be checked if a charge should be levied for late payment. This field is defaulted based on the customer's customer class. If you want to excuse a bill from late payment charge processing uncheck this box.
- **Late Payment Charge Date** is the date on which the system will levy a late payment charge if payment is not received. This date is calculated when the bill is completed. It is equal to the bill's due date plus the account's customer class's grace days. If this date does not fall on a workday, the date is set to the next available workday when the bill is completed. If you want to manually delay the onset of late payment charge processing, extend this date into the future.

Fastpath: Late payment charges. Late payment charges are calculated on the due date plus the grace period. Refer to How Late Payment Charges Get Calculated for more information.

Bill Summary summarizes the financial impact of the bill. The information that appears differs depending on whether the account is a balance-forward or open-item account. If the account is a balance-forward customer, the following information appears:

- **Previous Period's Balance** is the balance from the account's last bill.
- **Total Payments** is the total amount of frozen or canceled payment segment financial transactions linked to this bill. Frozen payment segments appear as negative numbers (decreasing the amount owed by the customer), while canceled payment segments appear as positive numbers (increasing the amount owed). You can drill down on this balance to view the financial transactions that contribute to this total. Refer to How To Remove Unwanted Payments From A Completed Bill for a description of how to perform common maintenance functions.
- **Total Adjustments** is the total amount of frozen or canceled adjustment financial transactions linked to this bill. You can drill down on the balance to view the financial transactions that contribute to this total. Refer to How To Remove Unwanted Adjustments From A Completed Bill for a description of how to perform common maintenance functions.
- **Total Bill Corrections** is the total amount of canceled and / or rebilled bill segment financial transactions linked to this bill. You can drill down on the balance to view the financial transactions that contribute to this total.
- **Total Current Billing Charges** is the total amount of frozen bill segment financial transactions linked to this bill. You can drill down on the balance to view the financial transactions that contribute to this total.
- **Ending Balance For This Period** is the sum of the five amounts above. It is the amount owed by the customer as of the end of the bill period.

If the account is an open-item customer, the following information appears under **Bill Summary**:

- **New Charges** is the total amount of frozen bill segment financial transactions linked to this bill. You can drill down on this value to view the financial transactions that contribute to this total.
- **Adjustments** is the total amount of frozen or canceled adjustment financial transactions linked to this bill. You can drill down on this value to view the financial transactions that contribute to this total. Refer to How To Remove Unwanted Adjustments From A Completed Bill for a description of how to perform common maintenance functions.
- **Corrections** is the total amount of canceled and / or rebilled bill segment financial transactions linked to this bill. You can drill down on this value to view the financial transactions that contribute to this total.
- **Total** is the sum of the above amounts. You can drill down on this value to view the financial transactions that contribute to this total.
- **Adjustments Presented on the Bill** - This link appears when one or more adjustments are presented on the bill, but does not affect the bill balance. On clicking the link, the **Financial Transactions on Bill** screen appears where you can view the financial transactions of those adjustments which presented on the bill.

If the account is an open-item customer, the following information appears under **Match Summary** (note, this section is not displayed for balance-forward accounts):

- X **Balanced Item(s)** contains the count and total amount of financial transactions linked to this bill that are linked to `balanced` match events.
- X **Unbalanced Item(s)** contains the count and total amount of financial transactions linked to this bill that are linked to `unbalanced` match events.

Copyright © 2024, Oracle and/or its affiliates.

- X **Disputed Item(s)** contains the count and total amount of financial transactions linked to this bill that are linked to `disputed` match events.
- X **Unmatched Item(s)** contains the count and total amount of financial transactions linked to this bill that are not linked to any match event.

Total Changes After Completion only appears if bill segments that appeared on the original bill have been subsequently canceled or rebilled. The amount displayed is the net effect of all cancels and rebills. Note, you can use the **Remarks** column in the bill segments grid (to the right) to see which bill segments have been cancelled / rebilled.

If 25 or fewer bill segments are linked to this bill, the grid in the lower right portion of the page contains one row for every bill segment linked to the bill.

The following points describe each column in the grid:

- The **Bill Segment** column contains a concatenation of each bill segment's division, contract type, status and bill period. The division and contract type come from the bill segment's contract.

  Note:

  If the multi parameter based pricing feature is enabled in the Banking module, the **Bill Segment** column displays the following:

  - Bill Segment Period
  - Product and product parameters for which the bill segment is created

  However, if the multi parameter based pricing feature is disabled in the Banking module, the **Bill Segment** column displays the following:

  - Bill Segment Period
  - Product and variance parameter for which the bill segment is created

- The **Current Amount** column contains the bill segment's effect on the contract's current balance.
- The **Status** column shows the bill segment's status. Refer to Bill Segment Lifecycle for the possible values.
- The **Remarks** highlight special situations. The following remarks may appear:
  - `Rebill after completion` appears if the bill segment is a rebill that was created after the bill was sent to the customer. The financial impact of this bill segment appears as a "bill correction" on the next bill produced for the customer.
  - `Rebill prior completion` appears if the bill segment is a rebill that was created before the bill was sent to the customer.
  - `Canceled after completion` appears if the bill segment was canceled after the bill was sent to the customer. The financial impact of this bill segment appears as a "bill correction" on the next bill produced for the customer.
  - `Canceled prior completion` appears if the bill segment was canceled before the bill was sent to the customer.
  - If the bill segment's status is `Error` or `Incomplete`, this column contains an *estimated* value of this bill segment. The estimated value is calculated by computing an average daily amount from the contract's `frozen` bill segments in the last 365 days. This average is then applied to the number of days in this bill segment's bill period. If the bill segment doesn't have a bill period (rare, but possible), the historical average is not displayed.
  - If the bill segment's status is `Error`, this column also contains the error message.
  - `Rebilled due to correction` appears if a correction note was generated for the bill segment. This remark appears only if you have the Correction Note Billing Option specified in the Installation record.
  - `Rebilled due to correction. Canceled after completion` appears for a correction note that spawns another correction note. This remark appears only if you have the Correction Note Billing Option specified in the Installation record.

If more than 25 bill segments are linked to this bill, the grid in the lower right portion of the page contains a summary of the various bill segments linked to the bill.

The following points describe each column in the grid:

Copyright © 2024, Oracle and/or its affiliates.

- The **Status** column shows the bill segment's status. Refer to Bill Segment Lifecycle for the possible values. Click on the hyperlink to transfer to the Bill Segments tab where the associated bill segments can be viewed.
- The **Total Bill Segments** column contains the number of bill segments linked to the bill with this **Status**. If the status is `Error` or `Incomplete`, this column contains an *estimated* value of these bill segments. The estimated value is calculated by computing an average daily amount from `frozen` bill segments in the last 365 days for each contract. This average is then applied to the number of days in each bill segment's bill period. Please note, an historical average cannot be computed if an `Error` or `Incomplete` bill segment does not have a bill period (rare, but possible).
- The **Total Amount** column contains the sum of the current amount of these bill segments.

At the bottom of the scroll is shown the **Total Generated Charge**. This represents the total amount of the bill segments. This amount may differ from the Total Current Billing Charges when:

- The bill is not `Complete`. A bill segment's financial effect is not shown on the bill until the bill is completed.
- A bill segment has been canceled / rebilled since the bill was completed.

The topics that follow describe each of the actions that appear in the **Bill Segment Action** and **Bill Action** areas. Refer to How To for a description of typical business processes that use these buttons.

Generate

The **Generate** button causes a bill segment to be created for each contract linked to the account. The system generates bill segments in the order of the billing processing sequence on each contract's contract type.

This button is enabled when you are in add mode (i.e., you are not displaying an existing bill) and you have selected an **Account ID**.

When clicked, the **Generate** window opens.

You must specify the following parameters in the **Generate** window to generate the bill segments:

- **Cutoff Date** is the last possible day of each bill segments bill period. The current date defaults when the window opens. Refer to Ways To Control The End Date Of A Bill Segment for more information.
- **Accounting Date** defines the financial period to which the new bill segments' financial transactions are booked. The current date defaults when the window opens.
- Check **Allow Estimates** if the system is allowed to estimate.
- Billing Option controls how the system retrieves the charges on the bill. Select the Use Cut Off Date option if the charges exists on the cut-off date.

After specifying the parameters, click **Calculate** to create a new bill and new bill segments for the account. You see a summary of the bill segments in the lower right portion of the page.

Freeze

Clicking **Freeze** causes all `freezable` bill segments to become `frozen`. Refer to Bill Segment Lifecycle for more information about freezing.

This button is enabled if:

- the `Freeze At Will` **Bill Segment Freeze Option** on the installation option has been selected AND
- `freezable` bill segments exist AND
- there are no `error`, `incomplete` or `pending cancel` bill segments.

> **Note: Freezing a day or more after generation.** If, during freezing, bill segments in `closed` accounting periods are detected, a pop-up appears in which you can specify a new accounting date; this date is updated onto the offending bill segments. This only happens if the accounting period closes before you freeze the bill segments (the accounting date is stamped on a bill segment when it's generated).

If, after freezing, you're ready to send the bill to the customer click Complete to finalize the bill.

Copyright © 2024, Oracle and/or its affiliates.

> **Note: If problems are detected after freezing.** A bill segment may not be changed after it is `frozen`. All subsequent changes must occur by canceling the frozen bill segment and creating a new bill segment. Refer to How To Cancel A Bill Segment and How To Cancel / Rebill A Bill Segment for more information.

Cancel Frozen

Clicking **Cancel Frozen** causes all `frozen` bill segments to become `canceled`. Refer to Bill Segment Lifecycle for more information about cancellation. This button is disabled if the bill is written off.

> **Note: Pending cancels are not affected.** Clicking **Cancel Frozen** does not affect bill segments that are `pending cancel`. To make a `pending cancel` bill segment `canceled`, transfer to Bill Segment - Main and click Cancel.

If you need to cancel a specific bill segment, either transfer to the next tab and cancel the bill segment in question or follow the instructions under How To Cancel A Bill Segment.

This button is enabled if `frozen` bill segments are linked to the bill.

When clicked, the **Cancel Frozen Bill Segments** window opens.

In the **Cancel Frozen Bill Segments** window, you must specify the following parameters to cancel the frozen bill segments:

- **Cancel Reason** defines why you are performing the cancellation.
- **Accounting Date** defines the financial period to which the canceled bill segments' financial transactions are booked. The current date defaults when the window opens.

After specifying the parameters, click **Calculate** to cancel the frozen bill segments.

> **Note: There is no Undo.** Unlike cancellations performed on Bill Segment - Main, there is no Undo action. This transaction moves the pending cancel bill segments to the cancel state immediately; whereas the bill segment transaction lets you examine the cancellation before you commit it. If you cancelled bill segments by mistake, you must generate and freeze new bill segments.

If the bill is `pending`, the cancellation causes the bill segments to be suppressed on the version of the bill sent to the customer (but they remain in the database for audit and financial reporting purposes).

If the bill is `complete` and you do not Reopen the bill, the financial impact of the canceled bill segments appears on the next bill created for the account (under Bill Corrections).

If the bill is `complete` you may be able to Reopen the bill and then Complete it. By doing this, you can suppress a `frozen` bill segment on a previously completed bill. This means that if you catch a problem after completion you don't necessarily have to show the problem to the customer. Rather, cancel the problem, reopen and then recomplete the bill (when you recomplete the bill the system marks the bill segment to be suppressed on the version sent to the customer because its cancellation appears on the same bill as the original charges).

Complete

The **Complete** button causes a bill to be finalized and marked for transmission to the customer. Refer to Bill Lifecycle for information about what happens during bill completion.

This button is enabled if:

- the `Freeze At Will` Bill Segment Freeze Option on the installation option has been selected AND
- the bill is `pending` AND
- there are no bill segments or there are bill segments and all are `frozen` or `canceled`.

When clicked, the **Complete** window opens.

If the **User May Override Bill Date** option has been enabled on the Installation Record, you may override the **Bill Date** prior to completion. Otherwise, the **Bill Date** is the current date and cannot be changed. The current date defaults when the window opens. If completing a correction note, the user is prompted to enter a **Correction Note Date instead of a bill date** in the **Complete** window.

Copyright © 2024, Oracle and/or its affiliates.

After specifying the above parameters, click **Calculate** to complete the bill.

Delete

The **Delete** button deletes a bill and its bill segments.

This button is enabled if:

- the bill is `pending` AND
- all bill segments are `incomplete`, `freezable` or in `error`.

Reopen

The **Reopen** button causes a bill to be reopened. Refer to Bill Lifecycle for information about why you would reopen a bill.

This button is enabled if:

- the bill is `complete` AND
- the bill is not written off AND
- this is the most recent bill produced for the account AND
- reopening hasn't been prohibited for the bill. Refer to Bill Lifecycle for information about conditions that can prevent a bill from reopening.

After reopening a bill, it returns to the `pending` state and you can make changes to the account's financial transactions (e.g., add payments, adjustments, and bill segments). After which, you click Complete or Freeze / Complete to finalize the bill. Only one of these buttons is shown - the specific one depends on how your organization has set the **Bill Segment Freeze Option** on the installation record.

Credit Note

The **Credit Note** button is used to cancel ALL bill segments and create a new bill that contains the cancellation details. We refer to this "negative bill" as a credit note. Refer to Credit Notes - Presenting Cancellation Details on a Separate Bill for more information.

This button is enabled if:

- the **Use Credit Notes** option on the Installation record is turned on AND
- the bill is `complete` AND
- the bill is not written off.

When clicked, the **Credit Note** window opens.

You must specify the following parameters in the **Credit Note** window to cancel the frozen bill segments:

- **Cancel Reason** defines why you are performing the cancellation.
- **Accounting Date** defines the financial period to which the canceled bill segments' financial transactions are booked. The current date defaults when the window opens.
- **Credit Note Date** defines the date of the credit note on which the cancellation details appear. The current date defaults when the window opens.

After specifying the parameters, click **Calculate** to cancel the frozen bill segments and create a credit note.

It should be noted that if this operation is successful, the new "negative bill" is `complete`. It is ready for transmission to the customer and for interface to the general ledger.

Note: If you want to issue a credit note for a specific bill segment. Rather than cancel all bill segments on a bill, you can create a credit note for a specific bill segment on a bill. To do this, drill down on the bill segment in question and then click the Credit Note button on the bill segment page. You can also issue a credit note for selected bill segments on the next tab by selecting the desired bill segments and clicking the **Credit Note** button.

Copyright © 2024, Oracle and/or its affiliates.

Correction Note

The **Correction Note** button is used to cancel and rebill ALL bill segments and create a new bill that contains the cancellation and rebill details. We refer to the resultant bill as a correction note. Refer to Correction Notes - Presenting Correction Notes on a Separate Bill for more information.

This button is enabled if:

- the **Correction Note** Billing Option is specified on the Installation record AND
- the bill is `complete` AND
- there is at least one frozen bill segment on the bill or all of the bill's bill segments are frozen or cancelled AND
- there are no bill segment financial transactions that have been transferred out of the system.

When clicked, the **Correction Note** window opens.

You must specify the following parameters in the **Correction Note** window to correct the bill segment:

- **Cancel Reason** defines why you are performing the correction.
- **Cutoff Date** is the last possible day of each bill period for a bill segment. The current date defaults when the window opens. Refer to Ways To Control The End Date Of A Bill Segment for more information.
- **Accounting Date** defines the financial period that the financial transactions are booked for a bill segment. The current date defaults when the window opens.
- Check **Allow Estimates** if the system is allowed to estimate consumption.
- **Billing Option** controls how the system retrieves the charges on the bill. Select the **Use Cut Off Date** option if the charges exists on the cut-off date.
- If you prefer to have the system use the consumption details used on the original bill segments, check **Use Old Read/Item/SQ**. If this switch is on, neither **Allow Estimates** nor **Billing Option** is relevant.
- After specifying the parameters, click **Calculate** to correct the bill segments and create a correction note. The Bill Page opens and displays the Correction Note.

Once you complete a correction note, it cannot be reopened.

After you create a correction note, the **Undo Correction Note** button is activated.

The button is enabled if:

- the **Correction Note** Billing Option is specified on the Installation record AND
- the bill is a correction note AND
- all of the bill segments for the correction note are freezable.

Note: If you want to issue a correction note for a specific bill segment. You can create a correction note for a specific bill segment on a bill. To do this, drill down on the bill segment in question and then click the **Correction Note** button on the bill segment page. You can also issue a correction note for selected bill segments by selecting the desired bill segments and clicking the **Correction Note** button.

Bill - Bill Segments

You can use this page to view all or selected bill segments linked to a bill. In addition, you can also perform certain maintenance functions on this page (see the details below).

Open **Billing Management, Bill, Bill Segments** to view this information.

Note: If the bill has more than 25 bill segments and you don't use the go to buttons in the bill segment grid on the Main tab to drill over to this information, the search criteria are intentionally left blank in order to avoid retrieving all bill segments (with the resultant slow response times). You must therefore use the **Contract Filter** and the **Status Filter** to define the type of bill segments that should be retrieved. See the `Description of page` below for more information about this page's search criteria.

Copyright © 2024, Oracle and/or its affiliates.

The `Description of page` section below describes the fields on this page. Refer to How To for a description of how to perform common maintenance functions.

Description of page

Bill Info contains a concatenation of important information about the bill (e.g., the bill date, its status, due date, ending balance, etc.).

Bill ID is the system-assigned unique identifier of the bill.

> **Note:** If the bill has more than 25 bill segments and you don't use the hyperlinks in the bill segment grid on the Main tab to drill over to this page, the **Filters** are intentionally left blank in order to avoid retrieving all bill segments (with the resultant slow response times). You must therefore use the **Contract Filter** and the **Status Filter** to define the type of bill segments that should be retrieved.

The area beneath **Bill Info** provides you with options that control which bill segments appear in the grid. The following points describe the various options:

- Use the **Contract Filter** to define the types of contracts whose bill segments appear in the grid. The following options are available:
  - `All.` Use this option if you do not wish to restrict bill segments based on contract attributes.
  - `Contract Type.` Use this option to restrict bill segments to those whose contracts are linked to a given **Division** and **Contract Type**.
- Use **Status Filter** to restrict the bill segments based on their status. The following options are available:
  - `All`. This option shows all bill segments regardless of status.
  - Refer to Bill Segment Lifecycle for the various status values.

Don't forget to click the search button after changing the filters.

The **Select All**/ **Unselect All** buttons are used to select bill segments if you plan on issuing any of the mass update actions at the bottom of the page. These buttons are enabled if either of the following conditions is true:

- All bill segments that appear *in the grid* are in the `Error`, `Freezable` and/or `Incomplete` states.
- All bill segments that appear *in the grid* are in the `Frozen` state

Refer to the description of the "mass update" actions below for more information.

> **Note: 100 bill segments at a time.** Clicking **Select All** selects the first 100 bill segments in the grid. If more than 100 bill segments exist, you must select them in batches.

The grid that follows contains the bill segments that match your search criteria. The following information appears in the grid:

- Select box. See the mass update description immediately above for conditions under which this checkbox can be used to select bill segments for mass update actions.
- The **Bill Segment** column contains a concatenation of the bill segment's division, contract type, status and bill period. The division and contract type come from the bill segment's contract. Click the hyperlink to transfer to Bill Segment - Main Information. On this page you can perform maintenance functions (e.g., cancel/rebill, delete, cancel, etc., depending on the segment's status) on the bill segment in question.
- The **Current Amount** column contains the bill segment's effect on the contract's current balance.
- The **Status** shows the bill segment's status. Refer to Bill Segment Lifecycle for the possible values.
- The **Remarks** column highlights special circumstances associated with the bill segment.
  - `Rebill after completion` appears if the bill segment is a rebill that was created after the bill was sent to the customer. The financial impact of this bill segment appears as a "bill correction" on the next bill produced for the customer.
  - `Rebill prior completion` appears if the bill segment is a rebill that was created before the bill was sent to the customer.

Copyright © 2024, Oracle and/or its affiliates.

  - `Canceled after completion` appears if the bill segment was canceled after the bill was sent to the customer. The financial impact of this bill segment appears as a "bill correction" on the next bill produced for the customer.
  - `Canceled prior completion` appears if the bill segment was canceled before the bill was sent to the customer.
  - If the bill segment's status is `Error` or `Incomplete`, this column contains an *estimated* value of this bill segment. The estimated value is calculated by computing an average daily amount from the contract's `frozen` bill segments in the last 365 days. This average is then applied to the number of days in this bill segment's bill period. If the bill segment doesn't have a bill period (rare, but possible), the historical average is not displayed.
  - If the bill segment's status is `Error`, this column also contains the error message.
  - `Rebilled due to correction note` appears if the correction note was generated for the bill segment. This remark is generated only if you have Correction Note specified in the Installation record.
  - `Rebilled due to correction. Canceled after completion` appears for a correction note that spawns another correction note. This remark is generated only if you have Correction Note specified in the Installation record.
- The **Service Quantity** and **UOM** (unit of measure) columns contain the amount of product's consumption that was billed. The UOM is defined on the contract's contract type.
- The **Bill Segment ID** is the system-assigned unique identifier of the bill segment.

This transaction has sophisticated logic that can be used to perform "mass updates" to the bill segments that appear in the grid (using the buttons at the bottom of the page). The following points describe these mass update actions:

Generate (Bill - Bill Segments)

The **Generate** button is used to delete and recreate one or more bill segments. This button is enabled if you select at least one row from the bill segments grid that's in the `Error`, `Freezable` and/or `Incomplete` state.

Note that you can only select a bill segment for regeneration if ALL bill segments *in the grid* are in the `Error`, `Freezable` and/or `Incomplete` states. Note, you can use the **Status Filter** to restrict the type of bill segments in the grid.

When clicked, the **Generate** window opens.

You must specify the following parameters in the **Generate** window to regenerate a new bill:

- **Accounting Date** defines the financial period to which the new bill segments' financial transactions will be booked. The current date defaults when the window opens.
- Check **Allow Estimates** if the system is allowed to estimate.
- Billing Option controls how the system retrieves the charges on the bill. Select the Use Cut Off Date option if the charges exists on the cut-off date.

After specifying the parameters, click **Calculate** to regenerate the selected bill segments. A summary of what the system does is presented next.

Freeze (Bill - Bill Segments)

The **Freeze** button is used to freeze one or more bill segments. This button is enabled if you select at least one row from the bill segments grid that's in the `Freezable` state AND the installation option indicates users are allowed to "freeze at will".

Note that you can only select a bill segment for freezing if ALL bill segments *in the grid* are in the `Freezable` state. Note, you can use the **Status Filter** to restrict the type of bill segments in the grid.

When clicked, the **Freeze** window opens.

Accounting Date is used to define the financial period to which the frozen bill segments' financial transactions are booked. The current date defaults when the window opens.

After specifying the above parameters, click **Freeze** to freeze the selected bill segments. A summary of what the system does is presented next.

Copyright © 2024, Oracle and/or its affiliates.

Delete (Bill - Bill Segments)

The **Delete** button is used to delete one or more bill segments. This button is enabled if you select at least one row from the bill segments grid that's in the `Error`, `Freezable` and/or `Incomplete` state.

Note that you can only select a bill segment for deletion if ALL bill segments *in the grid* are in the `Error`, `Freezable` and/or `Incomplete` states. Note, you can use the **Status Filter** to restrict the type of bill segments in the grid.

If the system is successful in deleting the selected bill segment(s), a confirmation window summarizes what the system just did (i.e., it shows the number of bill segments that were deleted).

Cancel/Rebill/Freeze (Bill - Bill Segments)

If problems are detected with a `frozen` bill segment, it should be canceled and a new bill segment should be created. We refer to this process as cancel / rebill. Refer to Cancel / Rebill Incorrect Bill Segments for more information.

Before you cancel / rebill, you'll probably need to fix the cause of the problem. The following table describes common problems and how to fix them.

Problem	**How To Fix It**
The wrong rate was used	Go to Contract - Rate Info and correct the rate.
A contract rider was applied incorrectly	Go to Contract - Rate Info and correct the contract rider information.
The wrong tax exemption amount was calculated	Go to Contract - Rate Info and correct the tax exemption information.

Note: Multi-Cancel/Rebill Saves Time. If the cause of the problem impacts many bill segments related to a contract, you should use the Multi Cancel Rebill transaction as it allows you to cancel / rebill an unlimited number of bill segments at once.

Note: Credit Notes. Using the cancel/rebill functionality causes the cancel and rebill details to be swept on to the customer's next bill unless you Reopen the bill. If your organization requires cancellation details to be presented to the customer separately from the new bill, you should not use the **Rebill** functionality. Rather, you should issue a Credit Note and then generate a new bill containing the revised charges.

The **Cancel/Rebill/Freeze** button is used to cancel, rebill and freeze one or more bill segments. This button is enabled if you select at least one row from the bill segments grid that's in the `Frozen` state AND the bill is not written off.

Note that you can only select a bill segment for cancel/rebill/freeze if ALL bill segments *in the grid* are in the `Frozen` state. Note, you can use the **Status Filter** to restrict the type of bill segments in the grid.

When clicked, the **Cancel/Rebill/Freeze** window opens.

- **Cancel Reason** defines why the bill segment(s) are being canceled.
- **Accounting Date** defines the financial period to which the canceled and new bill segments' financial transactions are booked. The current date defaults when the window opens.
- Check **Allow Estimates** if the system is allowed to estimate.
- Billing Option controls how the system retrieves the charges on the bill. Select the Use Cut Off Date option if the charges exists on the cut-off date.
- If you prefer to have the system use the consumption details used on the original bill segments, check **Use Old Read/Item/SQ**. If this switch is on, neither **Allow Estimates** nor **Billing Option** is relevant.

CAUTION: If the cause of the rebill is an incorrect rate and the new rate requires service quantities that were not calculated when the bill was originally calculated, do not turn on **Use Old Read/Item/SQ**. Why? Because you want the system to derive new service quantities during the rebill process.

After specifying the parameters, click **Calculate** to cancel, rebill and freeze the selected bill segments.

Copyright © 2024, Oracle and/or its affiliates.

> **Note: Errors.** Bill segment errors may occur when you cancel / rebill the bill segments. If this occurs, you must go to Bill Segment - Main for the `pending cancel` bill segment and use the **Undo** action to remove the offending bill segment (and restore the original segment to the `Frozen` state). Alternatively, you can drill down on the `Error` bill segment and use the **Regenerate** action after correcting the cause of the problem.

> **Note: There is no Undo.** Unlike cancel / rebills performed on Bill Segment - Main, there is no Undo action. This is because this transaction freezes the newly created bill segments after it cancel/rebills; whereas the bill segment transaction lets you examine the new bill segment before you freeze it. If you made a mistake, simply correct the cause of the mistake and then cancel / rebill again. The erroneous financial transactions are automatically suppressed on the next bill produced for the customer.

A confirmation window summarizes what the system just did (i.e., it shows the number of bill segments that were deleted / generated).

Cancel (Bill - Bill Segments)

The **Cancel** button is used to cancel one or more bill segments. This button is enabled if you select at least one row from the bill segments grid that's in the `Frozen` state AND the bill is not written off.

Note that you can only select a bill segment for cancellation if ALL bill segments *in the grid* are in the `Frozen` state. Note, you can use the **Status Filter** to restrict the type of bill segments in the grid.

When clicked, the **Cancel** window opens.

- **Cancel Reason** defines why the bill segment(s) are being canceled.
- **Accounting Date** is used to define the financial period to which the canceled bill segments' financial transactions are booked. The current date defaults when the window opens.

After specifying the above parameters, click **Calculate** to cancel the selected bill segments.

> **Note: There is no Undo.** Unlike cancellations performed on Bill Segment - Main, there is no Undo action. This transaction moves the pending cancel bill segments to the cancel state immediately; whereas the bill segment transaction lets you examine the cancellation before you commit it. If you cancelled bill segments by mistake, you must generate and freeze new bill segments.

Credit Note (Bill - Bill Segments)

The **Credit Note** button is used to cancel selected bill segments and create a new bill that contains the cancellation details. We refer to this "negative bill" as a credit note. Refer to Credit Notes - Presenting Cancellation Details on a Separate Bill for more information.

This button is enabled if:

- the **Use Credit Notes** option on the Installation record is turned on AND
- if you select at least one row from the bill segments grid that's in the `Frozen` state AND
- the bill is not written off.

Note that you can only select a bill segment for this action if ALL bill segments *in the grid* are in the `Frozen` state. Note, you can use the **Status Filter** to restrict the type of bill segments in the grid.

When clicked, the **Credit Note** window opens.

You must specify the following parameters in the **Credit Note** window to cancel the frozen bill segments:

- **Cancel Reason** defines why you are performing the cancellation.
- **Accounting Date** defines the financial period to which the canceled bill segments' financial transactions are booked. The current date defaults when the window opens.
- **Credit Note Date** defines the date of the credit note on which the cancellation details appear. The current date defaults when the window opens.

After specifying the parameters, click **Calculate** to cancel the frozen bill segments and create a credit note.

Copyright © 2024, Oracle and/or its affiliates.

It should be noted that if this operation is successful, the new "negative bill" is `complete`. Refer to Bill Lifecycle for information about the steps that take place during completion. It is ready for transmission to the customer and for interface to the general ledger.

> **Note: If you want to issue a credit note for an entire bill.** Rather than cancel a specific bill segment, functionality exists that allows you to create a credit note for ALL bill segments on a bill. To use this feature, display the bill in question and then click the Credit Note button on the bill page.

Bill - Bill Routings

This page is used to view the recipients of a bill. Refer to Printing Bills for a discussion of how this information is used to produce bill images.

This page is also used to request a new copy of a bill. Refer to How To Reprint A Bill (For The Original Recipients or For Someone New) for a description of how to do this.

> **Note: No routing information until completion.** A bill has no routing details until it is completed. At completion time, the system creates a routing detail for every person linked to the account that receives a copy of its bills. Refer to The Source Of Bill Routing Information for more information about how this information is constructed.

Use **Billing Management, Bill, Bill Routings** to open this page.

Description of Page

Bill Info contains a concatenation of important information about the bill (e.g., the bill date, its status, due date, ending balance). If sequential bill numbering has been enabled, the bill's number also appears.

Bill ID is the system-assigned unique identifier of the bill.

The **Bill Routing Information** scroll contains an entry for each recipient of the bill. This information is disabled after the bill has been routed to the recipient. The remainder of this discussion describes the fields that can be defined for each recipient of the bill.

Person ID identifies the person who was originally associated with the routing record. You may change the **Person ID** to any person in the system as this information is only used to default the recipient's name and address information, below.

The **Bill Routing Parameters** area contains information explaining how and when the bill is routed to the recipient. This information is gray when the bill routing information has been extracted or if a bill is reopened:

- **Bill Route Type** is the method used to transmit the bill to the customer. If the **Person ID** is linked to the account, the bill route type defaults from the person's Account - Person Information.
- **Sequence** is the system-assigned identifier assigned to this bill routing.
- **Batch Control** is the identifier of the batch of bills in which the recipient's copy was downloaded. The batch process is defined on the **Bill Route Type**.
- **Address ID** - Indicates the address to which the bills generated for the account should be sent.
- **Extract Date/Time** is the date and time the bill was downloaded. This information only appears after the download has commenced.
- **Format** indicates if the customer should receive a `Detailed` or a `Summary` bill. If the **Person ID** is linked to the account, this information defaults from Account - Person Information.
- **Reprint** is checked if a user created this routing. This switch is populated by the system and is always protected.
- Turn on **Do Not Extract** if the bill should not be routed to the individual for whatever reason.
- **Copies** indicates the number of copies of the bill the person receives. If the **Person ID** is linked to the account, this information defaults from Account - Person Information.
- **PO ID** is the purchase order Id the customer wants printed on their copy of the bill. If the **Person ID** is linked to the account, this information defaults from Account - Person Information.
- **Intercept** is the user ID of the individual who wants to review the bill before it is sent to the customer. This field is only populated when the account has a Bill Print Intercept specified. Intercept information is defined on Account - Main Information.

Copyright © 2024, Oracle and/or its affiliates.

The **Mailing Address Information** area contains the recipient's name and address.

- If the recipient has an *override mailing name*, **Name1**, **Name2** and **Name3** contain the person's override mailing name. Otherwise, **Name1** contains the recipient's primary name (this name might also include a prefix or suffix if the related Account - Person relationship has these fields defined).
- The remaining fields are the recipient's address. Refer to The Source Of Bill Routing Information for information about how the system constructs the recipient's address. Note:
  - If the recipient's address is a mailing address (as opposed to fax or email), the **Country** associated with the address controls the fields that appear in the address. Refer to Defining Countries for more information on the address constituents.
  - If a user manually adds a routing record, the mailing address information defaults from the information on the current routing record.
  - If a user specifies a **Person ID** that is linked to the bill's account, the **Bill Route Type** defined on the Account - Person record is defaulted.
  - If a user specifies a **Person ID** that is *not* linked to the bill's account, the **Bill Route Type** on the *installation record* is defaulted. Note, if the bill route indicates the bill is routed via the postal service, the system will only be able to default the person's address if the person has an override mailing address on their correspondence information.
  - If the user changes the **Bill Route Type** after the initial default, the system populates the address information as described under The Source Of Bill Routing Information.

Bill - Bill Messages

The Messages page is a grid containing one row for every message that appears on the bill. Open **Billing Management, Bill, Bill Messages** to view this information.

Note: No messages until completion. A bill has no messages until it is completed (unless you insert them manually). At completion time, the system assembles messages from the various message sources. Refer to The Source Of Bill Messages for information about these sources.

Note: The bill segment(s) may also have messages. Be aware that only account-oriented messages are linked to a bill. There may also be contract-oriented messages linked to the bill's bill segments. Refer to Bill - Bill Messages for information about the page on which contract-oriented messages are displayed. Refer to The Source Of Bill Messages for information about the various message sources and whether each is linked to a bill or a bill segment.

Note: Bill messages may contain dynamic information. Refer to Substituting Field Values Into A Bill Message for more information.

The **Description of Page** section below describes the fields on this page. Refer to How To Add Ad Hoc Messages To A Bill for a description of how to perform common maintenance functions.

Description of Page

Bill Info contains a concatenation of important information about the bill (e.g., the bill date, its status, due date, ending balance). If sequential bill numbering has been enabled, the bill's number also appears.

Bill ID is the system-assigned unique identifier of the bill.

The following columns are displayed in the grid:

- **Bill Message** is the code that identifies the bill message.
- **Message On Bill** is the message associated with the code.
- **Priority** is the bill messages priority (on the printed bill).
- **Insert Code** defines if the bill message causes an insert in the envelope.

Copyright © 2024, Oracle and/or its affiliates.

Bill - Characteristics

The Characteristics page is a grid containing one row for each characteristic linked to the bill. Bill characteristics are purely informational.

> **Note:** You can only choose characteristic types defined as permissible on a bill record. Refer to Setting Up Characteristic Types and Their Values for more information.

Description of Page

The following fields display on the grid:

- **Characteristic Type** must bea type of characteristic defined as permissible on a bill record.
- **Characteristic Value** indicates the value of the characteristic if the characteristic type is pre-defined or foreign key reference.

Bill - Auto Pay

You can use this page to view all or selected auto payments linked to a bill. In addition, you can also perform certain maintenance functions on this page (see the details below).

Open **Billing Management, Bill, Auto Pay** to view this information.

Bill Info contains a concatenation of important information about the bill (e.g., the bill date, its status, due date, ending balance, etc.).

Bill ID is the system-assigned unique identifier of the bill.

The **Select All**/ **Clear All** buttons are used to select automatic payments if you plan to stop the auto pay action at the bottom of the page.

Refer to the description of the stop auto pay action below for more information.

> **Note: 100 auto payments at a time.** Clicking **Select All** selects the first 100 auto payments in the grid. If more than 100 auto payments exist, you must select them in batches.

The grid that follows contains the auto payments for the searched **Bill ID**. The following information appears in the grid:

- Select box. See the **Stop Autopay** action enabled for the selected checkboxes.
- The **Auto Pay ID** indicates the automatic payments source through which the automatic payment is made.
- The **Settlement ID** indicates the settlement ID through which the automatic payment is settled.
- The **Settlement Account ID** indicates the account through which the automatic payment is settled.
- The **Auto Pay Creation Date** displays the date when automatic payment was created.
- The **Auto Pay Amount** displays the automatic payment amount.

  > **Note:** Installation Option Algorithm for Auto Pay Amount calculation should be set to APAM-SETT (CIPPKDSX).

- The **Auto Pay Stop User ID** indicates the user who has stopped the automatic payment.

  > **Note:** Data appears in this column when the automatic payment is stopped.

- The **Auto Pay Stop Date/Time** displays the date and time when automatic payment was stopped.

  > **Note:** Data appears in this column when the automatic payment is stopped.

- The **Auto Pay Stop Amount** displays the automatic payment amount which was stopped.

  > **Note:** Data appears in this column only when the automatic payment is stopped.

Copyright © 2024, Oracle and/or its affiliates.

You can stop automatic payments that appear in the grid (using the **Stop Autopay** button at the bottom of the page). The following point describes the auto pay action:

Stop Autopay (Bill - Auto Pay)

The **Stop Autopay** button is used to stop one or more automatic payment which is scheduled on extract date. This button is enabled if you select at least one row from the automatic payments grid

Click the **Stop Autopay** button, information regarding stopped automatic payment(s) appear in the grid.

- **Auto Pay Stop User ID** indicates the user who has stopped the automatic payment.
- **Auto Pay Stop Date/Time** displays the date and time when automatic payment was stopped.
- **Auto Pay Stop Amount** displays the automatic payment amount which was stopped.

How To

The topics in this section describe how to perform common bill maintenance functions.

How To Create A Bill For All Contracts Linked To An Account

99.9% of all bills are created by the batch bill process and require no human intervention before they are finalized and sent to a customer. The other 0.1% are created by users on-line / real time. This section explains how to create the 0.1%.

The following points describe how to use this page to create a new bill:

- Navigate to the Bill - Main Information page:
  - If you have already selected the customer (say on Control Central), select `Go To Bill +` option on the account context menu. This will transfer you to this page with the respective account populated.
  - If you haven't selected the account, use the main menu to navigate to this page (Main Menu, Billing Management, Bill, +) and then select the **Account ID** using the adjacent search button.
- Click the Generate button to create a new bill.
- You will see a summary of the bill segments in the lower right portion of the page. If there are errors, refer to How To Correct A Bill Segment That's In Error. If there are no errors, the **Bill Segment Freeze Option** on the installation option controls the next step:
  - If this option is set to `Freeze At Bill Completion`, click the Freeze / Complete button.
  - If this option is set to `Freeze At Will`, click the Freeze button and, if everything looks clean, click the Complete button.

How To Create A Bill For A Specific Contract

Most bills contain a bill segment for every contract linked to the account. If you want to produce a bill that contains bill segments for a subset of contracts (e.g., you want to create a one-time invoice for an account's billable charge contract), follow these steps:

- Navigate to the Bill - Main Information page:
  - If you have already selected the customer (say on Control Central), select `Go To Bill +` option on the account context menu. This will transfer you to this page with the respective account populated.
  - If you haven't selected the account, use the main menu to navigate to this page (Main Menu, Billing Management, Bill, +) and then select the **Account ID** using the adjacent search button.
- Click the save button to create a bill without bill segments for the account.
- Select `Go To Bill Segment +` option on the Bill's context menu to transfer to Bill Segment - Main.
- Select the **Contract ID** of the contract for which you want to create a bill segment.
- Click the Generate button to generate a bill segment.
- If an error exists, refer to How To Correct A Bill Segment That's In Error.
- If there are no errors, the **Bill Segment Freeze Option** on the installation option controls the next step:

Copyright © 2024, Oracle and/or its affiliates.

- If this option is set to `Freeze At Bill Completion`, return to Bill - Main (using the bill context menu) and click the Freeze / Complete button.
- If this option is set to `Freeze At Will`, click the Freeze button and, if everything looks clean, return to Bill - Main and click the Complete button.

How To Create a Bill with no Bill Segments

You may wish to generate a bill that contains no bill segments. For example, perhaps your customer is billed quarterly but pays monthly. During the months where no charges are calculated, you may choose to send a "bill" that does not show new charges, but includes the payments received. To produce a bill with no bill segments, follow these steps:

- Navigate to the Bill - Main Information page:
  - If you have already selected the customer (say on Control Central), select `Go To Bill +` option on the account context menu. This will transfer you to this page with the respective account populated.
  - If you haven't selected the account, use the main menu to navigate to this page (Main Menu, Billing Management, Bill, +) and then select the **Account ID** using the adjacent search button.
- Click the save button to create a bill without bill segments for the account.
- At this point either the **Complete** or **Freeze / Complete** button (based on the value of Bill Segment Freeze Option on installation options) becomes enabled. Click the applicable button for your installation to complete the bill. At this point, all the steps described for completing a bill in Bill Lifecycle are performed, including sweeping onto the bill any payments received since the last bill was generated.

How To Create A One-Time Invoice

The following points describe the steps necessary to create and bill a one-time invoice:

- If the customer to be invoiced doesn't already have an appropriate billable charge contract, create one. The easiest way to do this is from Start/Stop Service.
- Use Maintaining Billable Charges to add a billable charge for the contract. The billable charge contains the invoice lines and amounts. The system will produce the bill for this charge when the account's bill cycle next executes. If you can't wait until that time, proceed to the next step.
- If you created a new billable charge contract in step 1, you must activate it before billing can bill it. To do this, transfer to Contract - Main Information and click the activate button.
- Refer to How To Create A Bill For A Specific Contract for instructions describing how to create a bill with a bill segment for the billable charge contract.

How To Correct A Bill Segment That's In Error

A bill segment will exist in the `Error` state if a problem was encountered during the generation of a bill segment. To fix an `error` bill segment you must correct the cause of the error and then regenerate the bill segment. Refer to Bill Segment Errors for background information.

> **Note: Error segments created during batch billing.** If the `error` bill segment was created as part of the batch bill process, the system attempts to fix the offending segment(s) each night during the account's bill window by regenerating it. Therefore, if the cause of the error is fixed during the day, the system will automatically regenerate the bill segment when batch billing next runs; you don't have to manually correct each bill. And once a bill is error-free, it will be completed and sent to the customer. Refer to Batch Billing for more information.

Correcting the cause of a bill segment error can be difficult or trivial; we hope that the error messages (along with their long descriptions) provide a good description of how to resolve the issue.

After you have corrected the cause of the error, you should regenerate the bill segment. There are two ways to do this:

- You can use the Generate action on Bill - Bill Segment to regenerate one or more bill segments. We recommend using this method if the cause of the error affected many bill segments because you can regenerate many bill segments at a time on this page. Also, be aware that this transaction shows the estimated value of `error` bill segments so you can gauge whether to invest a lot of time in correcting the problem. For example, if the bill segment's estimated value is

Copyright © 2024, Oracle and/or its affiliates.

$10,000, it's probably worth correcting the problem before sending the bill out. However, if the bill segment's value is small, you may decide to simply delete the error bill segment and send the bill without it (the system will attempt to create a bill for the deleted period the next time the account is billed).

- You can use the Generate action on Bill Segment - Main to regenerate an individual bill segment. We'd recommend using this method if the problem is bill segment-specific and you need to examine the information that is snapshot on the bill segment to resolve the error (e.g., you need to look at the service quantities).

Note: Drilling Into Bill Segment Errors. When you use the bill segments in error To Do list to work through errors, you will be transferred to Bill Segment - Main with the bill segment displayed.

After all errors have been resolved on a bill, return to Bill - Main. The **Bill Segment Freeze Option** on the installation option controls the next step:

- If this option is set to `Freeze At Bill Completion`, click the Freeze / Complete button.
- If this option is set to `Freeze At Will`, click the Freeze button and, if everything looks clean, click the Complete button.

How To Correct A Bill That's In Error

Besides errors on bill segments, there may also be errors detected when the system attempts to complete a bill. For example, if the system cannot find a mailing address, the bill will be in error (as opposed to one of its bill segments). Refer to Bill Errors for background information.

To correct such an error:

- Navigate to Bill - Main.
- Look at the bill's error message. Push the view button to view the long explanation that will provide suggestions as to the cause of the error (and how to fix it). Transfer to the appropriate page to fix the problem.
- After fixing the cause of the error, return to Bill - Main. The **Bill Segment Freeze Option** on the installation option controls the next step:
- - If this option is set to `Freeze At Bill Completion`, click the Freeze / Complete button.
  - If this option is set to `Freeze At Will`, click the Freeze button and, if everything looks clean, click the Complete button.

How To Cancel A Bill Segment

If a bill segment was `frozen` and it should never have been created in the first place, the bill segment must be canceled. There are several ways to do this:

- You can use the Cancel Frozen action on Bill - Main to cancel all frozen bill segments linked to the bill. We recommend using this method if you need to cancel everything on a bill.
- You can use the Cancel action on Bill - Bill Segments to cancel selected bill segments. We recommend using this method if you need to cancel multiple bill segments.
- You can use the Cancel action on Bill Segment - Main to cancel an individual bill segment. We'd recommend using this method if you need to examine the information on the bill segment before canceling it.

If the related bill is `pending`, the cancellation will cause the bill segment to be suppressed on the version of the bill sent to the customer (but it remains in the database for audit and financial reporting purposes).

If the related bill is `complete` and you do not Reopen the bill, the financial impact of the canceled bill segment(s) will appear on the next bill created for the account (under Bill Corrections).

If the related bill is `complete` you may be able to Reopen the bill and then Complete it. By doing this, you can suppress a `frozen` bill segment on a previously completed bill. This means that if you catch a problem after completion you don't necessarily have to show the problem to the customer. Rather, cancel the problem, reopen and then recomplete the bill (when you recomplete the bill the system will mark the bill segment to be suppressed on the version sent to the customer because its cancellation appears on the same bill as the original charges).

Copyright © 2024, Oracle and/or its affiliates.

How To Cancel / Rebill A Bill Segment

If a bill segment was `frozen` but it contains incorrect information, it should be canceled and a new bill segment should be created. We refer to this process as cancel / rebill. Refer to Cancel / Rebill Incorrect Bill Segments for more information.

Before you cancel / rebill, you must correct the cause of the problem. The following table describes common problems and where to go to fix them.

Problem	How To Fix It
The wrong rate was used	Go to Contract - Rate Info and correct the rate.
A contract rider was applied incorrectly	Go to Contract - Rate Info and correct the contract rider information.
The wrong tax exemption amount was calculated	Go to Contract - Rate Info and correct the tax exemption information.

After correcting the cause of the problem, you're ready to cancel and rebill the offending bill segment(s). There are several ways to do this:

- You can use the Cancel / Rebill / Freeze action on Bill - Bill Segments to cancel, rebill and freeze selected bill segments. We recommend using this method if you need to cancel / rebill multiple bill segments due to a pervasive problem.
- You can use the Rebill action on Bill Segment - Main to cancel and rebill an individual bill segment. We'd recommend using this method if you need to examine the information on the bill segment before and after canceling it. If you use this method, you must also use the Freeze action to freeze the newly created bill segment.
- If the cause of the problem impacts many bill segments related to a contract, you should use the Multi Cancel Rebill transaction as it allows you to cancel / rebill an unlimited number of bill segments at once.

If the related bill is `pending`, the cancellation will cause the erroneous bill segment to be suppressed on the version of the bill sent to the customer (but it remains in the database for audit and financial reporting purposes).

If the related bill is `complete` and you do not Reopen the bill, the financial impact of the canceled bill segment(s) and their rebill bill segment(s) will appear on the next bill created for the account (under Bill Corrections).

If the related bill is `complete` you may be able to Reopen the bill and then Complete it. By doing this, you can suppress an erroneous `frozen` bill segment on a previously completed bill. This means that if you catch a problem after completion you don't necessarily have to show the problem to the customer. Rather, cancel / rebill the problem, reopen and then recomplete the bill (when you recomplete the bill the system will mark the bill segment to be suppressed on the version sent to the customer because its cancellation appears on the same bill as the original charges).

How To Create A Credit Note

A credit note is used to cancel bills segments and present the results of this cancellation on a separate bill. This separate bill is referred to as a credit note. Refer to Credit Notes - Presenting Cancellation Details on a Separate Bill for more information.

Note: Installation option must be turned on. If your organization creates credit notes, be sure to select the Credit Note Billing Option on the Installation Options. If you don't select this option, the credit note buttons will not be enabled.

Before creating a credit note, you must first determine whether a single bill segment must be canceled or whether all bill segments for a bill must be canceled:

- If you need to issue a credit note for the entire bill, navigate to the Bill page and click the Credit Note button.
- If you need to issue a credit note for a specific bill segment, navigate to the Bill Segment page and click the Credit Note button.

Copyright © 2024, Oracle and/or its affiliates.

> **Note: Open-item customers**. If any of the bill segments are associated with an open-item customer class AND the original bill segment is an open item, the match event will be created and balanced with the original FT and the cancellation FT, assuming you have the appropriate FT freeze algorithm plugged-in on the customer class.

> **Note: Issuing a credit note for a specific bill segment**. You may also issue a credit note for an individual bill segment. When you create a credit note for an individual bill segment, that bill segment is canceled and a new credit note bill is created for that single bill segment's cancellation information.

How To Override Service Quantities On A Specific Bill Segment

The calculation of the amount of the products used by a customer is one of the first steps that happens when a bill segment is created. 99.9% of the time, the system does a perfect job in this effort. However, there are rare circumstances when, due to missing or incorrect data, the quantity calculated by the system is not correct.

To help you deal with these rare situations, a "back door" exists that allows a bill segment's service quantities to be overridden. This truly is a "back door" because the proper way to fix the amount of a commodity is to correct the cause of the error and then cancel / rebill the bill segment (or issue a credit note).

This section describes how to override the total amount of service quantity billed on a bill segment.

- Display the `error` or `freezable` bill segment on Bill Segment - Main.

> **Note: Important!** You can only override consumption on a bill segment that is in `Error` or `Freezable`. This means you must generate a bill segment before you can override the SQ details.

- Transfer to the Bill Segment - SQ Details tab.
- Insert one row in the service quantity (SQ) grid (click the + button) for each service quantity you need to add. You may also change service quantities that were populated by the system when it initially generated the bill segment by overwriting the respective values.
- Save the bill segment.
- Generate the bill segment.
- If the bill segment still looks wrong, repeat the above points until the bill segment is correct.
- The **Bill Segment Freeze Option** on the installation option controls the next step:
  - If this option is set to `Freeze At Bill Completion`, return to Bill - Main (using the bill context menu) and click the Freeze / Complete button.
  - If this option is set to `Freeze At Will`, click the Freeze button and, if everything looks clean, return to Bill - Main and click the Complete button.

How To Remove Unwanted Adjustments (or Payments) From A Completed Bill

If the system (or a user) has created adjustments that have been swept onto a recently completed bill but you don't want them to appear on the bill (e.g., the system generated a late payment charge but you want to forgive it) perform the following steps:

- Cancel the adjustment (Adjustments - Main Information).
- Navigate to the Bill - Main page.
- Click the Reopen button to reopen the previously completed bill.
- Click the Complete button to recomplete the bill. When the bill is recompleted, the system will see that an adjustment that was swept onto the bill when it was originally completed has been canceled and it will mark the adjustment for suppression on the printed bill.

You can remove unwanted payments using an analogous approach - Cancel the payment, Reopen the bill, Complete the bill.

> **Note: Cannot reopen historical bills.** You may only reopen an account's most recent bill because recompleting the bill causes the ending-balance to change, and we don't want this to happen to historical bills.

Copyright © 2024, Oracle and/or its affiliates.

How To Add An Adjustment (or Payment) To A Completed Bill

If you want to add an adjustment to a completed bill perform the following steps:

- Add and freeze the adjustment (Adjustments - Main Information).
- Navigate to the Bill - Main page.
- Click the Reopen button to reopen the previously completed bill.
- Click the Complete button to recomplete the bill. When the bill is recompleted, the system will sweep the recently created financial transactions onto itself.

You can add a payment to a bill using an analogous approach - Add the payment, Reopen the bill, Complete the bill.

Note: Cannot reopen historical bills. You may only reopen an account's most recent bill because recompleting the bill causes the ending-balance to change, and we don't want this to happen to historical bills.

How To Reprint A Bill (For The Original Recipients or For Someone New)

A copy of a bill is sent to every individual / business specified in the bill's routing details. The system creates the routing details when a bill is completed. Refer to The Source Of Bill Routing Information for more information about how this information is compiled during bill completion.

If you want to reprint a bill OR if you want to send a bill to someone other than the original recipient(s):

- Navigate to the Bill - Main page.
- Transfer to the Bill - Bill Routings tab.
- Insert a new row (click the + button). Note well, you can only insert information on a bill that is `Complete`.
- Specify the name and address to which the bill should be sent. If you want to send the bill to someone other than the original recipient, simply specify the individual's name and address. If the recipient is a person in the system, you can select their person ID and let the system default this information for you.
- Save the bill.

Note: If a bill is `Complete` but the print process hasn't yet executed, you don't have to insert a new routing row. Rather, you can simply change the name and address on the original routing details.

How To Add Ad Hoc Messages To A Bill

If you want to add a message to a completed bill:

- Navigate to the Bill - Main page.
- Transfer to the Bill - Bill Messages tab.
- Insert a new row (click the + button). Note, you can only insert a message on the most recent bill that is `Complete`.
- Specify the desired message code.
- Save the bill.
- Reprintthe bill as per How To Reprint A Bill (For The Original Recipients or For Someone New).

Fastpath: Refer to The Source Of Bill Messages for more information about the messages the system automatically links to a bill when the bill is completed.

How To Remove A Completed Bill

You can't physically remove a `completed` bill from the system. However, you can remove all financial transactions from the bill (and therefore end up with a zero value bill). You can also suppress the bill from being printed. To do this, perform the following steps:

- If the bill has bill segments, cancel each bill segment. Refer to How To Cancel A Bill Segment for instructions describing how to do this.
- If the bill has payment amounts or adjustment amounts that were entered incorrectly, navigate to the payment and/or adjustment pages and cancel the payments / adjustments.

Copyright © 2024, Oracle and/or its affiliates.

- Navigate to the Bill - Main page.
- Click the Reopen button to reopen the previously completed bill.
- Transfer to the Bill - Bill Routings tab. Turn on the Do Not Print switch for every routing entry.
- Click the Complete button to recomplete the bill. When the bill is recompleted, the system will sweep the canceled segments onto it and the net effect should be zero current charges.

Note: Cannot reopen historical bills. You may only reopen an account's most recent bill because recompleting the bill causes the ending balance to change, and we don't want this to happen to historical bills.

If the bill has payment amounts or adjustment amounts that are valid, but you don't want them to appear on this bill, navigate to the **Bill, Financial Details** page. Drill over to each respective financial transaction by clicking the drill down button. Clicking this button transfers you to Financial Transaction - Main. On this page, clear out the value of the Bill ID. Clearing out the Bill ID unlinks the financial transaction from the bill. When the system next completes a bill for the account, it will find the unbilled financial transaction and link it to the bill.

How To Display A Bill On-line

To display a bill on-line, you must first have installed software to render the bill in a format suitable for displaying. You must also configure the system to invoke this software; otherwise, you will get a message indicating that the service is not available. This option can only be configured by your technical staff.

The system can be configured to work with different bill production software, and it comes configured with the ability to display a bill on-line using BI Publisher, DOC1, or Crystal. With DOC1, the bills are presented in PDF format. To view the bills, you must have Adobe Acrobat Reader installed.

To view a bill on-line, click the **Display Bill** button.

How to Enable Audit Facility for Bill Reopen

If the **CI_BILL** table is configured for audit, the system will create an audit entry in the **CI_AUDIT** table when you reopen a bill. The following fields are audited when you reopen a bill - **Bill Date** (BILL_DT), **Bill Status** (BILL_STAT_FLG), and **Completion Date/Time** (COMPLETE_DTTM). On reopening a bill, the system deletes the bill date and completion date/time and changes the status of the bill from **Complete** to **Pending**. You can verify the audit entries created in the **CI_AUDIT** table either using the **Audit Query by User** or **Audit Query by Table/Field/Key** screen.

How to Re-evaluate Auto Pay Details of a Bill

Oracle Revenue Management and Billing provides the ability to derive the auto pay ID using which an automatic payment is created for a completed bill. The system derives the auto pay ID which is effective on the bill completion date and stores it corresponding to the record in the **CI_BILL_ACH** table. There might be situations wherein the auto pay instructions of an account changes after the bill completion. For example, on completing a bill, the system derived the auto pay ID 123 for the bill on 4th Jan 2023 and the end date of the auto pay ID 123 was changed on 5th Jan 2023. In such case, the system should check whether the auto pay ID 123 is still effective for the bill. Until now, the system was not capable to re-evaluate the auto pay details of a completed bill.

Now, Oracle Revenue Management and Billing enables you to re-evaluate the auto pay ID of all completed bills whose records are present in the **CI_BILL_ACH** table. If you add or edit the non-rule based auto pay instructions (i.e. regular or split auto pay instruction) or rule based auto pay instructions of an account, the system sets the **RE_EVAL_APAY_SW** flag corresponding to the record in the **CI_ACCT** table to **Y**. A new batch named **Re-evaluate latest Auto-Pay details (C1-REAPY)** is introduced in this release. You should execute this batch before executing the **Create Automatic Payments and Refunds (APAYCRET)** batch.

The **Re-evaluate latest Auto-Pay details (C1-REAPY)** batch is used to re-evaluate the auto pay details for completed bills of an account where the **RE_EVAL_APAY_SW** flag is set to **Y**. It considers the records with the debit amount in the **CI_BILL_ACH** table.

If an account has non-rule based auto pay details, the system will fetch the auto pay ID for the completed bill once again and store it corresponding to the record in the **CI_BILL_ACH** table. However, if an account has rule based auto pay

Copyright © 2024, Oracle and/or its affiliates.

instructions, the system will fetch the auto pay ID for the completed bill once again and store it corresponding to the records in the **CI_BILL_ACH** and **C1_FT_EXT** tables. Note that if the system is not able to derive the auto pay details for a completed bill of an account, the system will create a To Do using the **C1-REAPY** To Do type.

Financial Transactions On A Bill

Open **Report and Inquiry, Financial Transactions on Bill** to view the financial transactions on a bill.

Note: You can also open this page using the go to buttons that prefix the financial transaction summaries on Bill - Main.

Description of page

Bill Id is the system-assigned unique identifier of the bill.

Use **Presentment Bill ID** to search financial transactions which are presented on a particular bill.

Account ID is the bill's account.

The area beneath **Account ID** provides you with options that control which financial transactions appear in the grid. The following points describe the various options:

- Use the **Contract Filter** to define the types of contracts whose financial transactions appear in the grid. The following options are available:
  - `All.` Use this option if you do not wish to restrict financial transactions based on contract attributes.
  - `Contract ID`. Use this option to restrict financial transactions to those of a specific contract.
  - `Contract Type.` Use this option to restrict financial transactions to those whose contracts are linked to a given **Division** and **Contract Type**.
- Use **FT Type Filter** to restrict the type of financial transactions that appear in the grid. The following options are available:
  - `Adjustments`. This option shows all financial transactions associated with adjustments.
  - `All`. This option shows all financial transactions.
  - `Corrections`. This option shows all financial transactions associated with corrections (i.e., cancel / rebills that have occurred after the bill was completed).
  - `Current Charges`. This option shows all financial transactions associated with bill segments (that aren't `Corrections`).
  - `Payments`. This option shows all financial transactions associated with payments.
- Use **Match Event Status Filter** to restrict the financial transactions based on the status of their match event. This filter only appears if the bill's account is an open-item customer. The following options are available:
  - `All`. This option shows all financial transactions.
  - `Balanced`. This option shows all financial transactions whose match event is `balanced`.
  - `Disputed`. This option shows all financial transactions whose match event is `disputed`.
  - `Unbalanced`. This option shows all financial transactions whose match event is `unbalanced`.
  - `Unmatched`. This option shows all financial transactions that are not linked to a match event.

CAUTION: Don't forget to click the search button after changing the filters or after selecting a new Bill ID.

The grid that follows contains the financial transactions that match your search criteria. The following information is displayed:

- **Match Event Status** shows the status of the financial transaction's match event. This column only appears if the bill's account is an open-item customer.
- **FT Type** displays the type of financial transaction except for adjustments. For adjustments, the adjustment type's description is shown. Click on the hyperlink to transfer to Financial Transaction - Main. On this page, you can change certain aspects of the FT in question. Refer to How To Remove Unwanted Adjustments (or Payments) From A Completed Bill for a description of how to perform common maintenance functions.

Copyright © 2024, Oracle and/or its affiliates.

- **Accounting Date** is the date the system uses to determine the financial transaction's accounting period in your general ledger.
- **Current Amount** contains the financial transaction's effect on the contract's current balance.
- **Payoff Amount** contains the financial transaction's effect on the contract's payoff balance. The **Payoff Amount** will be dim if it equals the **Current Amount**.
- **Show on Bill** indicates if information about the financial transaction appears on the customer's bill.
- **Contract Information** contains a summary of the respective contract.
- **Financial Transaction ID** is the system-assigned unique identifier of the financial transaction.
- **Bill ID** - Indicates the bill on which the financial transaction is created.
- **Presentment Bill ID** - Indicates the bill on which the financial transaction is presented.

At the bottom of the page is a summary of the financial transactions that match the search criteria.

Maintaining Bill Segments

A bill typically contains one bill segment for every active contract linked to its account. A bill segment contains information showing how the segment was calculated and how it should be printed on the customer's bill.

The actions taken to create a bill segment are dependent on the contract's contract type. The contract's rate is applied to the amount consumed to calculate how much the customer owes for the service. A bill segment and a financial transaction are generated to reflect the results of this calculation. Refer to How Rates Affect The Information On Bill Segments for the details. Refer to Effective Dates & Price Proration for information about how the system prorates changes to rates and prices during a bill period.

Note: Contract type controls everything. It's important to be aware that the system has other bill segment creation algorithms. For example, if the contract charges for a deposit, the system neither amalgamates consumption nor does it apply a rate. You may be wondering how the system knows which method to use? It uses the bill segment type on the contract's contract type. Refer to Designing and Defining Bill Segment Types for a description of the various bill segment creation algorithms that are supported in the system.

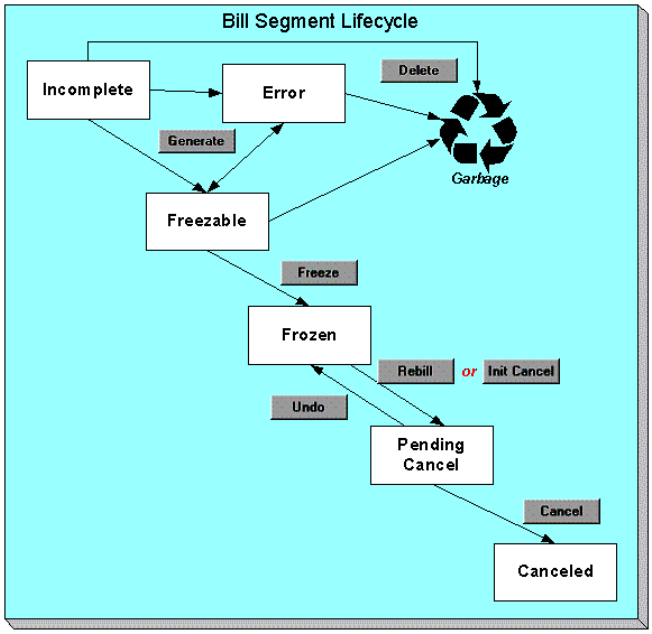
It's important to be aware that there are very few fields that are directly modifiable by a user. To modify most fields on a bill segment, you have to change source information (e.g., contract, rate) and then regenerate the bill segment. For example, if you want to change the rate used to calculate a bill segment, you must change the rate history on the respective contract and then cancel / rebill the bill segment. Refer to How To for step-by-step instructions that explain how to perform common bill segment maintenance functions.

Bill Segment Lifecycle

The following diagram shows the possible lifecycle of a bill segment.

CAUTION: This diagram only makes sense in the context of the page used to maintain bill segments. Refer to Bill Segment - Main Information for the details.

Copyright © 2024, Oracle and/or its affiliates.



Bill segments are initially created in the `Incomplete` state. Bill segments in this state don't have bill lines or a financial transaction; they are simply a stub awaiting generation.

Click **Generate** to generate the bill segment's bill lines and its financial transaction.

- If the system cannot generate the bill segment (for any number of reasons), the bill segment is moved to the `Error` state. You may regenerate the bill segment after correcting the source of the error (by clicking **Generate**). You may also delete such a bill segment.
- If the system successfully generates a bill segment, the bill segment becomes `Freezable`. If the bill segment is incorrect, correct the cause of the problem (e.g., change the contract's rate) and regenerate the bill segment (by clicking **Generate**).

Click the delete button to physically remove an `Incomplete`, `Error` or `Freezable` bill segment from the database. Note, batch billing will delete unfrozen bill segments when it attempts to create bill segments for an account that has unfrozen bill segments. Also note, if you've plugged-in the appropriate pre-completion algorithm, the system will delete bill segments in error on the last night of the bill cycle. Refer to DELBSEG for more information.

Click **Freeze** to freeze a bill segment and its financial transaction. After doing this, the bill segment's state becomes `Frozen` and the bill segment may now appear on a customer's bill. Also note, when a bill segment is frozen, its financial details will be interfaced to the general ledger the next time the GL interface executes.

You may not change a bill segment once it is frozen. However, you have two options if you want to change a frozen bill segment's financial impact:

- If you need to remove the financial effects of a bill segment, click **Init Cancel**. Clicking this button causes a new financial transaction to be generated and linked to the bill segment. This financial transaction reverses the financial effects of the original bill segment. The bill segment becomes `Pending Cancel` after the system successfully generates this financial transaction. At this point, you have two options:
  - Click **Cancel** to freeze the cancellation. Clicking this button causes the bill segment to move to the `Canceled` state. Once canceled, the bill segment cannot be uncanceled or changed.
  - Click **Undo** to return the pending cancellation to the `Frozen` state.
- Click **Rebill** if you need to recalculate the bill segment because something has changed since it was originally created (e.g., the contract's rate was changed, a new read was entered). Clicking this button causes the following events to take place:
  - The original bill segment is moved to the `Pending Cancel` state.
  - A new bill segment is created in the `Freezable` (or `Error`) state.
- At this point, you have two options.

Copyright © 2024, Oracle and/or its affiliates.

- Click **Freeze** to make the new bill segment `Frozen` and make the original bill segment `Canceled`.

Note: Multi-Cancel/Rebill Saves Time. If you need to rebill many bill segments related to a contract (e.g., because many bill segments were created using the wrong rate), you should use the Multi Cancel Rebill transaction as it allows you to cancel / rebill an unlimited number of bill segments at once.

Note: Using the cancel and cancel/rebill functionality will cause the cancel and rebill details to be swept on to the customer's next bill. If your organization requires cancellation details to be presented to the customer separately from the new bill, then you will need to use Credit Notes. Refer to Credit Notes - Presenting Cancellation Details on a Separate Bill for more information.

Click **Undo** to delete the new bill segment and return the original bill segment to the `Frozen` state.

Bill Segment - Main Information

Fastpath: The **Description of Page** section below describes the fields on this page. Refer to How To for a description of how to perform common bill maintenance functions.

The Main page contains core bill segment information. Open **Billing Management, Bill Segment, Main** to view this information.

Note: Most bill segments are created by a background process. Refer to A High Level Overview Of The Bill Creation Process for information describing how the system creates a bill segment.

Description of Page

Bill Seg Info is a concatenation of the bill segment's division, contract type, status, bill period and amount. **Bill Segment ID** is the system-assigned unique identifier of the bill segment.

Account ID references the bill segment's account.

Current Amount is the bill segment's effect on the contract's current balance. **Payoff Amount** is only shown if it differs from current amount. Refer to Billing - Current Balance versus Payoff Balance for more information.

Bill ID is the system-assigned unique identifier of the bill on which the bill segment appears. A concatenation of its bill date, status, due date and amount is displayed.

Contract ID contains information about the bill segment's contract. A concatenation of the contract's division, contract type, status and start date is displayed.

If the quotation module is not turned off, you can generate simulated bill segments when you create a quote. You can view simulated bill segments by drilling down on the desired bill segment from Quote - Quote Details. When a simulated bill segment is displayed on this page, the **Billing Scenario ID** that controlled the bill segment's billing period (and consumption) is displayed. Refer to Creating Quotes And Quote Details for more information.

Period is the start and end dates of the bill segment.

Bill Cycle displays the bill cycle and window start date of the bill segment's bill. This information is only populated on bills when they are generated on-schedule by the batch billing background process. Bills generated manually, i.e. off-schedule, do not have this information.

Status is the bill segment's status. Refer to Bill Segment Lifecycle for the potential values and how to handle a bill segment when it exists in a given state.

If the **Closing** switch is on, this is the final bill segment for the contract (i.e., the contract has been stopped and this is the final bill segment).

If the **SQ Override** switch is on, a user has overridden the service quantities. Refer to How To Override Service Quantities On A Specific Bill Segment for more information.

Create Dt/Tm is the date and time on which the bill segment was created or regenerated.

Copyright © 2024, Oracle and/or its affiliates.

In the **Message** area, a brief error message appears if there's a problem with the bill segment. The message area is suppressed if there are no problems with the bill.

Push the magnifying glass button to view the long explanation. The long explanation will provide information about the cause of the error (and how to fix it).

If the bill segment is a rebill of a canceled bill segment, a reference to the original bill segment is displayed. Click the adjacent go to button to transfer to the bill segment that was canceled.

If the bill segment was canceled and rebilled by another segment, a reference to the new bill segment is displayed. Click the adjacent go to button to transfer to the new bill segment that superceded the canceled segment.

The topics that follow describe each of the actions on that appear in the **Bill Segment Actions** area. Refer to How To for a description of typical business processes that use these buttons.

Note: Mass update actions. This page allows you to work on individual bill segments. If you want to update many bill segments linked to a bill at once, try using the mass update actions available on Bill -Bill Segments.

Generate (Bill Segment)

Click the **Generate** button to create a new a bill segment.

This button is enabled when either of the following conditions is true:

- You are in add mode (i.e., you are not displaying an existing bill segment) and you have selected an **Account ID**, **Bill ID** and **Contract ID**.
- You are displaying a bill segment that is `Incomplete`, `Error`, or `Freezable`. Refer to Bill Segment Lifecycle for more information about these status values.

When clicked, the **Generate** window opens.

You must specify the following parameters to generate a new bill segment:

- **Cut-off Date** is the last possible day of the bill segment's bill period. The current date defaults when the window opens. Refer to Ways To Control The End Date Of A Bill Segment for more information.
- **Accounting Date** is used to define the financial period to which the new bill segment's financial transaction will be booked. The current date defaults when the window opens.
- Turn on **Allow Estimates** if the system is allowed to estimate.
- Billing Option controls how the system retrieves the charges on the bill. Select the Use Cut Off Date option if the charges exists on the cut-off date.

After specifying the above parameters, click the **Calculate** button to create a new bill segment for the bill.

If an error exists, refer to How To Correct A Bill Segment That's In Error.

If there are no errors, the **Bill Segment Freeze Option** on the installation option controls the next step:

- If this option is set to `Freeze At Bill Completion`, return to Bill - Main (using the bill context menu) and click the Freeze / Complete button.
- If this option is set to `Freeze At Will`, click the Freeze button and, if everything looks clean, return to Bill - Main and click the Complete button.

Delete (Bill Segment)

The **Delete** button deletes a bill segment.

This button is enabled if you are displaying a bill segment that is `Incomplete`, `Error`, or `Freezable`. Refer to Bill Segment Lifecycle for more information about these status values.

Freeze (Bill Segment)

Clicking the **Freeze** button causes a `freezable` bill segment to become `frozen`. Refer to Bill Segment Lifecycle for more information about freezing.

Copyright © 2024, Oracle and/or its affiliates.

This button is enabled if:

- the `Freeze At Will` **Bill Segment Freeze Option** on the installation option has been selected AND
- the bill segment is `freezable`.

Note: Freezing a day or more after generation. If the bill segment is linked to a `closed` accounting period, a pop-up appears in which you can specify a new accounting date. This will only happen if the accounting period closes before you freeze the bill segment (the accounting date is stamped on a bill segment when it's initially generated).

Note: If problems are detected after freezing. A bill segment may not be changed after it is `frozen`. All subsequent changes must occur by canceling the frozen bill segment and creating a new bill segment. Refer to How To Cancel A Bill Segment and How To Cancel / Rebill A Bill Segment for more information.

Rebill (Bill Segment)

If problems are detected with a `frozen` bill segment, it should be canceled and a new bill segment should be created. We refer to this process as cancel / rebill. Refer to Cancel / Rebill Incorrect Bill Segments for more information.

Before you cancel / rebill, you'll probably need to fix the cause of the problem. The following table describes common problems and where to go to fix them.

Problem	How To Fix It
The wrong rate was used	Go to Contract - Rate Info and correct the rate.
A contract rider was applied incorrectly	Go to Contract - Rate Info and correct the contract rider information.
The wrong tax exemption amount was calculated	Go to Contract - Rate Info and correct the tax exemption information.

Note: Multi-Cancel/Rebill Saves Time. If the cause of the problem impacts many bill segments related to a contract, you should use the Multi Cancel Rebill transaction as it allows you to cancel / rebill an unlimited number of bill segments at once.

Note: Credit Notes. Using the cancel/rebill functionality will cause the cancel and rebill details to be swept on to the customer's next bill unless you Reopen the bill. If your organization requires cancellation details to be presented to the customer separately from the new bill, you should not use the **Rebill** functionality. Rather, you should issue a Credit Note and then generate a new bill containing the revised charges.

The **Rebill** button causes an existing bill segment to be canceled, and a new bill segment to be created.

This button is enabled when you display a `Frozen` bill segment AND the bill segment's bill is not written off. Refer to Bill Segment Lifecycle for more information about these status values.

When clicked, the **Rebill** window opens.

You must specify the following parameters to cancel and rebill a new bill segment:

- **Cancel Reason** defines why the bill segment is being canceled.
- **Cutoff Date** is the last day of the new bill segment. The cutoff date on the bill segment defaults when the window opens.
- **Accounting Date** is used to define the financial period to which the canceled and new bill segments' financial transactions will be booked. The current date defaults when the window opens.
- Turn on **Allow Estimates** if the system is allowed to estimate.
- Billing Option controls how the system retrieves the charges on the bill. Select the Use Cut Off Date option if the charges exists on the cut-off date.
- Turn on **Use Old Read/Item/SQ for Regen** if you want the system to use the product consumption details on the original bill segment when it calculates the new bill segment.

Click the **Calculate** button to cancel the original bill segment and create a new bill segment.

Copyright © 2024, Oracle and/or its affiliates.

At this point, there are now two bill segments - the original bill segment is in the state of `Pending Cancel`, the new bill segment is in the `Freezable` (or `Error`) state. If you want to back out, click the go to button to return to the original bill segment and then click Undo (this will delete the new bill segment and return the original bill segment to the `Frozen` state). If the new bill segment looks correct, click Freeze to move the original bill segment to the `Canceled` state and the new bill segment to the `Frozen` state.

Init Cancel (Bill Segment)

The **Init Cancel** button causes the first step of the bill segment cancellation process to be executed. You'd click this button if a `frozen` bill segment should never have been created (i.e., you want to remove the financial impact of a bill segment from a customer's balance).

This button is enabled when you display a `Frozen` bill segment AND the bill segment's bill is not written off. Refer to Bill Segment Lifecycle for more information about these status values.

When clicked, the **Cancel** window opens.

You must specify the following parameters to cancel and rebill a new bill segment:

- **Cancel Reason** defines why the bill segment is being canceled.
- **Accounting Date** is used to define the financial period to which the canceled bill segment's financial transactions will be booked. The current date defaults when the window opens.

Click the **Calculate** button to cancel the original bill segment.

At this point, the bill segment is in the state of `Pending Cancel`. If you want to back out, click Undo; the bill segment will be returned to the `Frozen` state. Click Cancel to move the bill segment to the `Canceled` state.

> **Note: Credit Notes**. Using the cancel functionality will cause the cancel details to be swept on to the customer's next bill unless you Reopen the bill. If your organization requires cancellation details to be presented to the customer separately from the new bill, you should not use the cancelfunctionality. Rather, you should issue a Credit Note and then generate a new bill containing the revised charges.

Undo (Bill Segment)

The **Undo** button returns a `Pending Cancel` bill segment to the `Frozen` state. If a `Pending Cancel` bill segment was created during a Cancel / Rebill, the newly created `Freezable` bill segment is also deleted.

This button is enabled when you display a `Pending Cancel` bill segment. Refer to Bill Segment Lifecycle for more information about this status.

Cancel (Bill Segment)

Clicking the **Cancel** button causes a `Pending Cancel` bill segment to become `Canceled`. Refer to Bill Segment Lifecycle for more information about cancellation.

This button is enabled if a `Pending Cancel` bill segment is displayed AND the bill segment's bill is not written off.

When clicked, the **Cancel** window opens.

You must specify the following parameters to cancel the frozen bill segments:

- **Cancel Reason** defines why you are performing the cancellation.
- **Accounting Date** defines the financial period to which the canceled bill segments' financial transactions will be booked. The current date defaults when the window opens.

After specifying the above parameters, click the **Calculate** button to cancel the frozen bill segments.

If the related bill is `pending`, the cancellation will cause the bill segment to be suppressed on the version of the bill sent to the customer (but it remains in the database for audit and financial reporting purposes).

If the related bill is `complete` and you do not Reopen the bill, the financial impact of the canceled bill segment will appear on the next bill created for the account (under Bill Corrections).

Copyright © 2024, Oracle and/or its affiliates.

If the related bill is `complete` you may be able to Reopen the bill and then Complete it. By doing this, you can suppress a `frozen` bill segment on a previously completed bill. This means that if you catch a problem after completion you don't necessarily have to show the problem to the customer. Rather, cancel the problem, reopen and then recomplete the bill (when you recomplete the bill the system will mark the bill segment to be suppressed on the version sent to the customer because its cancellation appears on the same bill as the original charges).

Credit Note (Bill Segment)

The **Credit Note** button is used to cancel a bill segment and create a new bill that contains the cancellation details. We refer to this "negative bill" as a credit note. Refer to Credit Notes - Presenting Cancellation Details on a Separate Bill for more information.

> **Note: If you want to issue a credit note for an entire bill.** Rather than cancel a specific bill segment, functionality exists that allows you to create a credit note for ALL bill segments on a bill. To use this feature, display the bill in question and then click the **Credit Note** button on the bill page.

This button is enabled if:

- the **Use Credit Notes** option on the Installation record is turned on AND
- the bill segment is `frozen` AND
- the bill segment's bill is not written off.

When clicked, the **Credit Note** window opens.

You must specify the following parameters to cancel the `frozen` bill segment:

- **Cancel Reason** defines why you are performing the cancellation.
- **Accounting Date** defines the financial period to which the canceled bill segment's financial transaction will be booked. The current date defaults when the window opens.
- **Credit Note Date** defines the date of the credit note on which the cancellation details will appear. The current date defaults when the window opens.

After specifying the above parameters, click the **Calculate** button to cancel the frozen bill segment and create a credit note.

It should be noted that if this operation is successful, the new "negative bill" will be `complete` (meaning that it will be ready for transmission to the customer and for interface to the general ledger).

Bill Segment - SQ Details

> **Fastpath:** The **Description of Page** section below describes the fields on this page. Refer to How To Override Service Quantities On A Specific Bill Segment for instructions describing how to override this information.

The SQ Details page contains information about the service quantities (SQ) that will be priced by the contract's rate. Open **Billing Management, Bill Segment, SQ Details** to view this information.

Description of Page

Bill Seg Info is a concatenation of the bill segment's division, contract type, status, bill period and amount. **Bill Segment ID** is the system-assigned unique identifier of the bill segment.

The **Service Quantity** details grid is a snapshot of the service quantities amassed from:

- Those items for which the system estimates product consumption.
- The SQ rules defined on the contract's rate.
- The register rules defined on the contract's rate.
- Billable charges that supply service quantities.

One row exists for every unique combination of unit of measure (UOM), variance parameter, and service quantity identifier (SQI) associated with the contract. Refer to *UOM versus Variance Parameter versus SQI* for more information.

Copyright © 2024, Oracle and/or its affiliates.

This information may be overridden on a bill segment in the `Error` or `Freezable` state. Insert one row in the service quantity grid (click the + button) for each UOM/Variance Parameter/SQI you need to add. You may also change amounts that were populated by the system when it initially generated the bill by simply overwriting the information.

CAUTION: The proper way to fix the SQ details on a bill segment is to correct the cause of the error. Overriding the SQ details on a bill segment is a last resort that should only be used when you can't fix the cause of the problem. Refer to How To Cancel / Rebill A Bill Segment for ways to fix common problems.

The following information is displayed in the grid:

- **Unit of Measure** is the unit of measure of the service quantity.
- **Variance Parameter** is the variance parameter of the service quantity.
- **SQI** is the service quantity identifier of the service quantity.
- **Initial Service Quantity** is the initial quantity amassed by the system before application of the rate's SQ rules.
- **Billable Service Quantity** is the service quantity that will be priced by the contract's rate. This amount differs from the initial service quantity when the rate's SQ rules manipulate product's consumption.

Bill Segment - Calc Lines

Fastpath: The **Description of Page** section below describes the fields on this page. Refer to How To Cancel / Rebill A Bill Segment for instructions describing how to regenerate this information if the bill segment is incorrect.

The Calc Lines page contains information about the bill calculation lines produced when the system generates the bill segment. These lines are the source of the details printed on the customer's bill. Open **Billing Management, Bill Segment, Calc Lines** to view this information.

Fastpath: Most contracts use a rate to calculate the information that appears on this page. Refer to How Rates Affect The Information On Bill Segments for more information.

Description of Page

Bill Seg Info is a concatenation of the bill segment's division, contract type, status, bill period and amount. **Bill Segment ID** is the system-assigned unique identifier of the bill segment.

The Bill Segment Header (**Bill Seg Hdr)** scroll contains one entry for every version of the rate that was used to calculate the bill segment's lines. Because most bill segments use a single rate version, you typically only see one entry in the scroll.

The following information is displayed above the bill calculation line grid.

- **Sequence** will be 1 unless multiple versions of the rate were in effect during the bill period. If there was more than one effective rate version, there will be a separate set of calc lines for each version applied. The lines that apply to the first part of the bill period would have a Sequence of 1, the lines for the next part of the bill period would have a Sequence of 2 and so on.
- **Start Date - End Date** is the portion of the bill segment period that the calculation details apply.
- **Amount** is the sum of the bill calculation lines in the grid.
- **Billable Charge ID** - Indicates the billable charge for which the bill segment is generated. In addition, this field has a context menu which helps in navigating to the **Billable Charge** screen.
- **Desc on Bill** is summary information about the calculation details that will be printed on the customer's bill.
- **Rate Version** is the ID of the rate version and its effective date. This information only appears if a rate was used to calculate the charges.

The calculation lines grid is a snapshot of how the system calculated the bill segment amount. One row exists for every calculation involved in this process. This information is for audit purposes only and may not be modified. The following information is displayed in the grid:

- An extra column with a drill down button appears if the bill calculation line is the result of a variance parameter pricing rate component that contains an audit algorithm.

Copyright © 2024, Oracle and/or its affiliates.

- A drill down button appears in the **Calc Line Char** column if the bill calculation line has characteristics. The Char column only appears if at least one of the calculation lines has a characteristic.
- **Sequence** is the system-assigned unique identifier of the calculation detail row.
- **Description on Bill** is the information about the bill line that appears on the customer's bill.
- **Calculated Amount** is the calculated amount associated with the bill line.
- The **Print** switch controls whether information about this line will print on the customer's bill.
- The **Appears in Summary** switch defines if this line's amount also appears on a summary line. This switch plays a part at bill print time - those lines that appear in a summary print in the left dollar column, those that don't appear in a summary print in the right dollar column.
- The system automatically populates this switch as follows:
  - Bill segments created by applying a rate have this switch turned on if the corresponding rate component is summarized on a summary rate component
  - Bill segments created from billable charges use the value of this switch as defined on the billable charge. Refer to Maintaining Billable Charges for more information.
- **Unit of Measure** (UOM) is the unit of measure of the service quantity priced on the calculation line.
- **Variance Parameter** of the service quantity priced on the calculation line.
- **SQI** is the service quantity identifier of the service quantity priced on the calculation line.
- **Billable Service Quantity** is the service quantity priced on the calculation line. This quantity will be different from the measured product's consumption is there are SQ rules or register rules in effect.
- **Base Amount** is used by calculation lines (e.g. taxes) that are cross-referenced to other calculation lines and whose value(s), therefore, depend on the amounts calculated by those other lines. The Base Amount shows the total amount derived from the cross-referenced line(s) that the current line then used to calculate its billed amount.
- **Rate Comp** onent **Seq** uencerefers to the sequence number of the rate component on the applicable rate version that was used to calculate the line.
- **Measures Peak Qty** is checked if the UOM priced on the calculation line is used to measure a peak quantity.
- **Exempt Amount** is the amount of the calculated charge that the customer doesn't have to pay because they are tax exempt.
- **Distribution Code** is the distribution code associated with the calculation line. This distribution code is used to build the general ledger details on the bill segment's financial transaction.
- **Description** describes the characteristic value that was used when the line's amount was calculated. This information is only displayed if the line was calculated using a bill factor (because only bill factors use characteristic values). Refer to *An Illustration Of A Bill Factor And Its Characteristics* for more information.

Bill Segment - Financial Details

Fastpath: The **Description of Page** section below describes the fields on this page. Refer to How To Cancel / Rebill A Bill Segment for instructions describing how to regenerate this information if it is incorrect.

The Financial Details page contains information about the financial effects of the bill segment. Open **Billing Management, Bill Segment, Financial Details** to view this information.

Note: Refer to The Source Of GL Accounts On A Bill Segment's Financial Transaction for how the system compiles this information.

Description of Page

Bill Seg Info is a concatenation of the bill segment's division, contract type, status, bill period and amount. **Bill Segment ID** is the system-assigned unique identifier of the bill segment.

The **Bill Segment FT** scroll contains one entry for every financial transaction (FT) associated with the bill segment. All bill segments have an FT that shows the impact of the bill segment on the general ledger. A bill segment may have an additional FT (for a maximum of two) if it is cancelled. The second FT shows the impact of the cancellation on the general ledger.

Copyright © 2024, Oracle and/or its affiliates.

The following financial transaction information is displayed above the general ledger distribution grid.

- **FT Type** displays the type of financial transaction (FT).
- **Show on bill** indicates if information about the FT appears on the customer's bill.
- **FT ID** is the system-assigned unique identifier of the FT. Click the go to button to transfer to the Financial Transaction - Main page. On this page you can change certain aspects of the FT in question. Refer to How To Remove Unwanted Adjustments (or Payments) From A Completed Bill for a description of how to perform common maintenance functions.
- The posted by information that appears adjacent to the FT ID shows the date on which the FT was frozen and the user ID of the user who performed the operation.
- The "Linked to bill on" message appears when the financial transaction is linked to a bill. A FT is linked to a bill the next time a bill is completed for the account. Therefore, if a user cancels a bill segment, the FT associated with the cancellation will not be linked to a bill until the next bill is completed (although the financial effect of the cancellation takes place immediately). Click the adjacent go to button to transfer to the bill on which the FT appears.
- **Current Amount** contains the FT's effect on the contract's current balance.
- **Payoff Amount** contains the FT's effect on the contract's payoff balance. This field is only visible when the payoff amount differs from the current amount.
- **Accounting Date** is the date the system uses to determine the FT's accounting period in your general ledger.
- **Arrears Date** is the date the FT starts aging.

The journal details grid is a snapshot of the distribution codes used to generate the GL accounts that are affected by the FT. This information is for audit purposes only and may not be modified. The following information is displayed in the grid:

- **Seq** is the system-assigned unique identifier of the distribution line.
- **Distribution Code** is the distribution code used to generate the GL account affected by the distribution line. Drill over to the FT if you need to see the actual GL account.
- **Amount** is the amount debited / credited (credits are shown as negative values).
- **Statistics Code** and **Statistics Amount** are only specified when a statistical quantity is associated with the distribution line.
- **Description** describes the characteristic value that was used when the line's amount was calculated. This information is only displayed if the distribution line was derived from bill calculation lines that were calculated using a bill factor (because only bill factor's use characteristic values). Refer to *An Illustration Of A Bill Factor And Its Characteristics* for more information.
- If you have configured your installation options to indicate that fund accounting is `practiced`, the description of the **Fund** associated with this distribution code is displayed.

Bill Segment - Bill Segment Messages

The Messages page is a grid containing one row for every message that appears on the bill segment. Open **Billing Management, Bill Segment, Bill Segment Messages** to view this information.

Note: No messages until completion. A bill segment has no messages until it is completed (unless you insert them manually). At completion time, the system assembles messages from the various message sources. Refer to The Source Of Bill Messages for information about these sources.

Note: The bill may also have messages. Be aware that only contract-oriented messages are linked to a bill segment. There may also be account-oriented messages linked to the bill. Refer to Bill - Bill Messages for information about the page on which account-oriented messages are displayed. Refer to The Source Of Bill Messages for information about the various message sources and whether each is linked to a bill or a bill segment.

Description of Page

Bill Seg Info is a concatenation of the bill segment's division, contract type, status, bill period and amount. **Bill Segment ID** is the system-assigned unique identifier of the bill segment.

Copyright © 2024, Oracle and/or its affiliates.

The following columns are displayed in the grid:

- **Msg Code** is the code that identifies the bill message.
- **Message On Bill** is the message associated with the code.
- **Priority** is the bill messages priority (on the printed bill).
- **Insert Code** defines if the bill message causes an insert in the envelope.

Bill Segment - Sub Bill Segments

The Sub Bill Segments page contains the sub bill segments that are associated with the master contract's bill segment. Open this page using **Billing Management, Bill Segment, Sub Bill Segment**.

Description of Page

Bill Seg Info is a concatenation of the bill segment's division, contract type, status, bill period and amount. **Bill Segment ID** is the system-assigned unique identifier of the bill segment.

The following information is displayed in the grid.

- Click the adjacent **Go To** button to transfer to the bill segment.
- The **Bill Segment** column contains a concatenation of the bill segment's business unit, contract type, status, bill period and amount. The business unit and contract type come from the bill segment's contract.
- The **Current Amount** column contains the bill segment's effect on the sub contract's current balance.
- The **Status** column contains the bill segment's current status.

Financial - Bill Segment Calculation Line Characteristics

This page displays the characteristics that are linked to a specific bill segment calculation line. The information on this page is for audit purposes only and cannot be changed.

Open this page by clicking the **Char** go to button on the calculation lines grid of the Bill Segment - Calc Lines page.

Description of Page

Account is the account to which the bill segment is linked. The name of the main person on the account appears next to the account ID.

The **Contract** is the contract to which the bill segment applies. Basic information about the contract appears adjacent.

The **Bill Seg ID** and **Bill Seg Header** displays the identification of the bill segment and header associated with the calculation line characteristic.

Calc Line displays the calculation line number with which the characteristics are associated.

The **Description on Bill** displays the description of the calculation line that appears on the customer's bill.

The characteristics grid displays the **Characteristic Type** s and the **Characteristic Value** s associated with the calculation line.

Fastpath: For more information about characteristics, refer to Setting Up Characteristic Types and Their Values.

Multi Cancel/Rebill

The Multi-Cancel/Rebill transaction is used to cancel / rebill (and freeze) one or more bill segments linked to a contract.

Note: You would typically use this transaction when something has been wrong for an extended period of time. For example, if the customer was assigned the wrong rate when they first started service, you would correct the customer's rate using Contract - Rate Info and then use this transaction to cancel / rebill the customer's historical bill segments.

Copyright © 2024, Oracle and/or its affiliates.

The following topics describe how to use this transaction.

Multi Cancel/Rebill - Main

The main page is used to select the bill segments to be cancel/rebilled and to initiate the cancel/rebill request. Open **Accounting and Receivables, Multi-Cancel/Rebill** to use this transaction.

Description of Page

This page contains all non-cancelled bill segments associated with a given **Contract ID**. The contract's **Contract Info**, **Account ID** (and name), characteristic **Location** (and address), **Current Balance** and **Payoff Balance** are displayed at the top of page.

When the page first opens, the grid is populated with ALL non-cancelled bill segments displayed in reverse chronology order. All `frozen` bill segments are pre-selected for cancel/rebill processing. The following information appears in the grid:

- Selected Switch. If this switch is turned on, the bill segment will be cancel / rebilled when the **Cancel/Rebill/Freeze** button is clicked. You can use the mouse to toggle the value of this switch on a specific bill segment. Alternatively, you can click the **Select All** button or **Unselect All** button to toggle the switch for all `frozen` bill segments. The checkbox is disabled if the bill segment's bill is written off. Note: the total number of selected bill segments is shown above the grid for confirmation purposes.
- **Start Date.** This is the first day of the bill segment.
- **End Date.** This is the last day of the bill segment.
- **Create Date/Time.** This is the day/time the bill segment was created.
- **Amount.** This is the payoff amount of the bill segment.
- **Original Amount.** This column only contains a value if the bill segment is a rebill of another bill segment. In this situation, one of the following values will be displayed:
  - If the amount of the new bill segment differs from the amount of the original bill segment, the amount of the original bill segment will be displayed.
  - If the amount of the new bill segment equals the amount of the original bill segment, the word `No change` will be displayed.
- **Bill Segment Information.** This is the standard bill segment summary information.

Note: Sorting By Values In The Grid. Just like all grids in the system, you can click on a column heading to resort the grid by the value of the respective column. This might be useful to sort the bill segments in **Amount** order.

After selecting the bill segments to be cancel/rebilled, click the **Cancel/Rebill/Freeze** button. When this button is clicked, an additional informational page displays. Specify the following information and then click **Calculate** to cancel the selected bill segment and create new bill segments:

- **Cancel Reason** defines why the bill segment(s) are being canceled.
- **Accounting Date** is used to define the financial period to which the canceled and new bill segments' financial transactions will be booked.
- Turn on **Allow Estimates** if the system is allowed to estimate consumption.
- Use the **Billing Option** to define how the system looks for meter reads:
  - Select `Use Schedule` if the system should select reads based on the contract's SP's service cycle schedule.
  - Select `Use Cut off Date` if the system should only select reads from the end date of the existing bill segment.
- If you'd prefer to have the system use the consumption details snapshot on the original bill segments, turn on **Use Old Read/Item/SQ**. If this switch is on then neither **Allow Estimates** nor **Billing Option** is relevant.

CAUTION: If the cause of the rebill is an incorrect rate and the new rate requires service quantities that were not calculated when the bill was originally calculated, do not turn on **Use Old Read/Item/SQ**. Why? Because you want the system to derive new service quantities during the rebill process.

Copyright © 2024, Oracle and/or its affiliates.

Note: The Accounting Date is defaulted to the current date and the Use Old Read/Item/SQ switch is turned on.

After specifying the above information, the system cancel / rebills each segment in chronological order (i.e., earlier segments will be cancel / rebilled before later segments). The new amount of each bill segment is shown in the **Original Amount** column. You can navigate to the Graph page to graphically view the financial ramifications of the cancel/rebills.

Note: Errors. It's possible that bill segment errors will occur when you cancel / rebill the bill segments. If this occurs, you must go to Bill Segment - Main for the `pending cancel` bill segment and use the **Undo** action to remove the offending bill segment (and restore the original segment to the `Frozen` state). Alternatively, you can drill down on the `Error` bill segment and use the **Regenerate** action after correcting the cause of the problem.

Note: There is no Undo. Unlike cancel / rebills performed on Bill Segment - Main, there is no Undo action. This is because the multi-cancel/rebill transaction freezes the newly created bill segments after it cancel/rebills; whereas the bill segment transaction lets you examine the new bill segment before you freeze it. If you made a mistake, simply correct the cause of the mistake and then cancel / rebill again. The erroneous financial transactions will be automatically suppressed on the next bill produced for the customer.

Multi Cancel/Rebill - Graph

This page contains a graph that shows the rebilled and original payoff amounts for a contract's non-cancelled bill segments. Open **Accounting and Receivable, Multi Cancel/Rebill** and navigate to the **Graph** page to view this information.

Description of Page

This page contains a graph that shows the payoff amounts for a contract's non-cancelled bill segments (represented by the blue bars in the graph). If a bill segment is a rebill of another bill segment, the payoff amount of the original bill segment is displayed on a red-colored bar adjacent to the respective blue bar.

Note: Click a bar. You can click on a bar to drill down to a bill segment.

Contract Billing History

Open **Report and Inquiry, Contract Billing History** to view a history of all bill segments produced for a contract.

Description of Page

The **Account ID** is displayed with the account's name adjacent.

The **Contract Information** shows a summary of information about the contract. The **Contract ID** is the unique identifier of this contract.

One row is displayed for every bill segment ever produced for the selected contract.

The following information is displayed for each bill segment:

- The **Start Date** and **End Date** of the bill segment.
- The number of **Days** in the bill segment period.
- The **Status** of the bill segment
- The **Current** and **Payoff Amount** s of the bill segment.
- The **UOM** (Unit of Measure) of the service quantity designated as the "graph UOM" for the contract's contract type.
- The total amount of the service **Billable Service Quantity** that was billed.
- The **Average Daily Service Quantity** is the **Billable Service Quantity** divided by the numbers of days of service.

If you need to see more detailed information about the bill segment, click the go to button to transfer to the bill segment page.

Copyright © 2024, Oracle and/or its affiliates.

Bill Exception

If errors are detected during the billing process, it may cause a record to be written to the bill exception table with a message indicating the nature of the severe error.

To view the messages associated with the exception records, schedule the TD-BIERR background process. This process generates a To Do entry for every record in the bill exception table.

Fastpath: Refer to How to Correct a Bill That's In Error for instructions describing how to correct a bill.

Bill Segment Exception

If errors are detected during the billing process, it may cause a record to be written to the bill segment exception table with a message indicating the nature of the severe error.

To view the messages associated with the exception records, schedule the TD-BSERR background process. This process generates a To Do entry for every record in the bill segment exception table.

Fastpath: Refer to How To Correct A Bill Segment That's In Error for instructions describing how to correct a bill segment.

Invoice Frequency

Oracle Revenue Management and Billing enables you to create recurring billable segments from a billable charge at the set invoice frequency (such as weekly, monthly, quarterly, and so on). The **Invoice Frequency** screen allows you to define, edit, and delete an invoice frequency. It consists of the following zones:

- Invoice Frequency List on page 242
- Invoice Frequency on page 243

Invoice Frequency List

The **Invoice Frequency List** zone lists invoice frequencies that are already defined in the system. It contains the following columns:

Column Name	Column Description
Invoice Frequency	Displays the invoice frequency.
Description	Displays the description of the invoice frequency.
Daily Frequency	Indicates the frequency in days.
Monthly Frequency	Indicates the frequency in months.

You can define an invoice frequency by clicking the **Add** link in the upper right corner of this zone. On clicking the **Broadcast** (📡) icon corresponding to the invoice frequency, the **Invoice Frequency** zone appears.

Related Topics

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
How to view the details of an invoice frequency	Viewing the Invoice Frequency Details on page 243
How to define an invoice frequency	Defining an Invoice Frequency on page 244

Invoice Frequency

The **Invoice Frequency** zone displays the details of the invoice frequency. It contains the following fields:

Field Name	Field Description
Invoice Frequency	Displays the invoice frequency.
Description	Displays the description of the invoice frequency.
Monthly Frequency	Indicates the frequency in months.
Daily Frequency	Indicates the frequency in days.

In addition, this zone contains the following buttons:

Button Name	Button Description
Edit	Used to edit the details of the invoice frequency.
Delete	Used to delete the invoice frequency.
	Note: You can only delete an invoice frequency which is not yet used.

By default, the **Invoice Frequency** zone does not appear in the **Invoice Frequency** screen. It appears when you click the **Broadcast** (ᯤ) icon corresponding to the invoice frequency in the **Invoice Frequency List** zone.

Related Topics

For more information on...	See...
How to edit an invoice frequency	Editing an Invoice Frequency on page 245
How to delete an invoice frequency	Deleting an Invoice Frequency on page 246

Viewing the Invoice Frequency Details

Procedure

To view the details of an invoice frequency:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **I** and then click **Invoice Frequency**.
   A sub-menu appears.
3. Click the **Search** option from the **Invoice Frequency** sub-menu.
   The **Invoice Frequency** screen appears.
4. In the **Invoice Frequency List** zone, click the **Broadcast** (ᯤ) icon corresponding to the invoice frequency whose details you want to view.
   The **Invoice Frequency** zone appears.

Copyright © 2024, Oracle and/or its affiliates.

5. View the details of the invoice frequency in the **Invoice Frequency** zone.

Related Topics

For more information on...	See...
Invoice Frequency screen	Invoice Frequency on page 242
Invoice Frequency List zone	Invoice Frequency List on page 242
Invoice Frequency zone	Invoice Frequency on page 243

Defining an Invoice Frequency

Procedure

To define an invoice frequency:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **I** and then click **Invoice Frequency**.
   A sub-menu appears.
3. Click the **Add** option from the **Invoice Frequency** sub-menu.

   The **Invoice Frequency** screen appears. It contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Invoice Frequency	Used to specify the invoice frequency.	Yes
Monthly Frequency	Used to specify the frequency in months. For example, if you want to define the invoice frequency as Monthly, enter `1` in this field. If you want to define the invoice frequency as Quarterly, enter `3` in this field. And, if you want to define the invoice frequency as Yearly, enter `12` in this field.	Yes (Conditional) **Note:** This field is required when you are defining a frequency in months.
Daily Frequency	Used to specify the frequency in days. For example, if you want to define the invoice frequency as Weekly, enter `7` in this field. If you want to define the invoice frequency as Fortnightly, enter `15` in this field.	Yes (Conditional) **Note:** This field is required when you are defining a frequency in days.
Description	Used to specify the description for the invoice frequency.	Yes

Tip: Alternatively, you can access the **Invoice Frequency** screen by clicking the **Add** link in the upper right corner of the **Invoice Frequency List** zone.

4. Enter the required details in the **Invoice Frequency** screen.
5. Click **Save**.
   The invoice frequency is defined.

Related Topics

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
Invoice Frequency List zone	Invoice Frequency List on page 242

Editing an Invoice Frequency

Procedure

To edit an invoice frequency:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **I** and then click **Invoice Frequency**.
   A sub-menu appears.
3. Click the **Search** option from the **Invoice Frequency** sub-menu.
   The **Invoice Frequency** screen appears.
4. In the **Invoice Frequency List** zone, click the **Broadcast** (ᯤ) icon corresponding to the invoice frequency whose details you want to edit.
   The **Invoice Frequency** zone appears.
5. Click **Edit**.

   The **Invoice Frequency** screen appears. It contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Invoice Frequency	Displays the invoice frequency.	Not applicable
Monthly Frequency	Used to specify the frequency in months. For example, if you want to define the invoice frequency as Monthly, enter `1` in this field. If you want to define the invoice frequency as Quarterly, enter `3` in this field. And, if you want to define the invoice frequency as Yearly, enter `12` in this field.	Yes (Conditional) **Note:** This field is required when you are defining a frequency in months.
Daily Frequency	Used to specify the frequency in days. For example, if you want to define the invoice frequency as Weekly, enter `7` in this field. If you want to define the invoice frequency as Fortnightly, enter `15` in this field.	Yes (Conditional) **Note:** This field is required when you are defining a frequency in days.
Description	Used to specify the description for the invoice frequency.	Yes

6. Modify the required details in the **Invoice Frequency** screen.
7. Click **Save**.
   The changes made to the invoice frequency are saved.

Related Topics

For more information on...	See...
Invoice Frequency screen	Invoice Frequency on page 242
Invoice Frequency List zone	Invoice Frequency List on page 242
Invoice Frequency zone	Invoice Frequency on page 243

Copyright © 2024, Oracle and/or its affiliates.

Deleting an Invoice Frequency

Procedure

To delete an invoice frequency:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **I** and then click **Invoice Frequency**.
   A sub-menu appears.
3. Click the **Search** option from the **Invoice Frequency** sub-menu.
   The **Invoice Frequency** screen appears.
4. In the **Invoice Frequency List** zone, click the **Broadcast** (ᯤ) icon corresponding to the invoice frequency that you want to delete.
   The **Invoice Frequency** zone appears.
5. Click **Delete**.
   A message appears confirming whether you want to delete the invoice frequency.

   Note: You can only delete an invoice frequency which is not yet used.
6. Click **OK**.
   The invoice frequency is deleted.

Related Topics

For more information on...	See...
Invoice Frequency screen	Invoice Frequency on page 242
Invoice Frequency List zone	Invoice Frequency List on page 242
Invoice Frequency zone	Invoice Frequency on page 243

Charge Type

Oracle Revenue Management and Billing enables you to indicate whether the billable charge is created for the premium, fees, down payment, taxes, or commission. This is possible through specifying the charge type for the billable charge. In other words, the charge type indicates the type of the billable charge. You can create both recurring and non recurring billable charges for all types of charges. However, you can specify a charge type while creating a billable charge for a contract when the charge type is linked to the contract type using which the contract is created. You can link a charge type to more than one contract type.

The **Charge Type** screen allows you to define, edit, and delete a charge type. It consists of the following zone:

- Charge Type List on page 246

Charge Type List

The **Charge Type List** zone lists the type of charges that are already defined in the system. It contains the following columns:

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
Charge Type	Displays the charge type.
Description	Displays the description of the charge type.
Charge Type Category	Indicates the category to which the charge type belongs.
Charge Type Sub Category	Indicates the subcategory to which the charge type belongs.
Edit	On clicking the **Edit** (✎) icon, the **Charge Type** screen appears where you can edit the details of the charge type.
Duplicate	On clicking the **Duplicate** (⧉) icon, the **Charge Type** screen appears where you can define a new charge type using an existing charge type.
Delete	On clicking the **Delete** (🗑) icon, you can delete the charge type.
	Note: You can delete a charge type which is not yet linked to any contract type.

Note: Pagination is used to display limited number of records in the **Search Results** section. You can use the navigation links, such as **Previous** and **Next** to navigate between pages.

On clicking the **Broadcast** (📡) icon corresponding to a charge type, the **Charge Type** zone appears with the details of the respective charge type.

You can filter the list using various search criteria (such as, **Charge Type**, **Category**, **Charge Type Description**, and **Charge Type Sub Category**) available in the **Filter** area. By default, the **Filter** area is hidden. You can view the **Filter** area by clicking the **Filters** (▽) icon in the upper right corner of this zone.

Related Topics

For more information on...	See...
How to edit a charge type	Editing a Charge Type on page 250
How to copy a charge type	Copying a Charge Type on page 251
How to delete a charge type	Deleting a Charge Type on page 252

Charge Type

The **Charge Type** zone displays the details of the charge type. It contains the following sections:

- **Main** - Displays basic information about the charge type. It contains the following fields:

Column Name	Column Description
Charge Type	Displays the charge type
Description	Displays the description of the charge type.
Charge Type Category	Indicates the category to which the charge type belongs. The valid values are: • Additional Taxes • Commission • Down Payment • Fees • Premium- Endorsement • Premium- Business

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
	• Tax
Charge Type Sub Category	Indicates the subcategory to which the charge type belongs.

- **Characteristics** - Lists the characteristics defined for the charge type. It contains the following columns:

Column Name	Column Description
Effective Date	Indicates the date from when the characteristic is effective for the charge type.
Characteristic Type	Indicates the characteristic type.
Characteristic Value	Displays the value of the characteristic type.

- **Record Actions** - This section contains the following buttons:

Button Name	Button Description
Edit	Used to edit the details of the charge type.
Delete	Used to delete the charge type.
	Note: You can delete a charge type which is not yet linked to any contract type.
Duplicate	Used to create a new charge type using an existing charge type.

- **Record Information** - This section contains the following field:

Field Name	Field Description
Business Object	Indicates the business object using which the charge type is created. In addition, a context menu appears corresponding to this field which helps in navigating to other screens in the application.
	Note: It has a link. On clicking the link, the **Business Object** screen appears where you can view the details of the respective business object.

By default, the **Charge Type** zone does not appear in the **Charge Type** screen. It appears when you click the **Broadcast** (📶) icon corresponding to a charge type in the **Charge Type List** zone.

Related Topics

For more information on...	See...
How to edit a charge type	Editing a Charge Type on page 250
How to copy a charge type	Copying a Charge Type on page 251
How to delete a charge type	Deleting a Charge Type on page 252

Defining a Charge Type

Prerequisites

To define a charge type, you should have:

- Values defined for the **CHG_TYPE_SUB_CAT** lookup field

Procedure

To define a charge type:

Copyright © 2024, Oracle and/or its affiliates.

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **C** and then click **Charge Type**.
   A sub-menu appears.
3. Click the **Add** option from the **Offset Request Type** sub-menu.

   The **Charge Type** screen appears. It contains the following sections:

   - **Main** - Used to specify basic details about the charge type.
   - **Characteristics** - Used to define the characteristics for the charge type.

   The **Main** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Charge Type	Used to specify the charge type.	Yes
Description	Used to specify the description for the charge type.	Yes
Charge Type Category	Used to indicate the category to which the charge type belongs. The valid values are: • Additional taxes • Commission • Down Payment • Fees • Premium- Endorsement • Premium- New Business • Tax	Yes
Charge Type Sub Category	Used to indicate the subcategory to which the charge type belongs.	No

4. Enter the required details in the **Main** screen.
5. Define characteristics for the charge type, if required.
6. Click **Save**.
   The charge type is defined.

Related Topics

For more information on...	See...
Charge Type screen	Charge Type on page 246
How to define a characteristic for a charge type	Defining a Characteristic for a Charge Type on page 249

Defining a Characteristic for a Charge Type

Prerequisites

To define a characteristic for a charge type, you should have:

- Characteristic types defined in the application (where the characteristic entity is set to **Charge Type**)

Procedure

To define a characteristic for a charge type:

1. Ensure that the **Characteristics** section is expanded when you *a*re defining, editing, or copying a charge type.

   The **Characteristics** section contains the following fields in a grid:

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
Effective Date	Used to specify the date from when the characteristic is effective for the charge type.	Yes (Conditional) **Note:** This field is required when you are defining a characteristic for the charge type.
Characteristic Type	Used to indicate the characteristic type. **Note:** The list includes only those characteristic types where the characteristic entity is set to **Charge Type**.	Yes (Conditional) **Note:** This field is required when you are defining a characteristic for the charge type.
Characteristic Value	Used to specify the value for the characteristic type. **Note:** If you select a predefined characteristic type, the **Search** (🔍) icon appears corresponding to the **Characteristic Value** field. On clicking the **Search** icon, the **Predefined Characteristic Search** window appears. On specifying the value for a predefined characteristic type, the description of the characteristic value appears corresponding to the **Characteristic Value** field.	Yes (Conditional) **Note:** This field is required when you are defining a characteristic for the charge type.

2. Enter the required details in the **Characteristics** section.
3. If you want to define more than one characteristic for the charge type, click the **Add** (+) icon and then repeat step 2.

> **Note:** However, if you want to remove a characteristic from the charge type, click the **Delete** (🗑) icon corresponding to the characteristic.

Related Topics

For more information on...	See...
How to define a charge type	Defining a Charge Type on page 248
How to edit a charge type	Editing a Charge Type on page 250
How to copy a charge type	Copying a Charge Type on page 251

Editing a Charge Type

Procedure

To edit a charge type:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **C** and then click **Charge Type**.
   The **Charge Type** screen appears.

Copyright © 2024, Oracle and/or its affiliates.

3. In the **Charge Types** zone, click the **Edit** (✎) icon in the **Edit** column corresponding to the charge type whose details you want to edit.

   The **Charge Type** screen appears. It contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Charge Type	Displays the charge type.	Not applicable
Description	Used to specify the description for the charge type.	Yes
Charge Type Category	Indicates the category to which the charge type belongs. The valid values are: • Additional taxes • Commission • Down Payment • Fees • Premium- Endorsement • Premium- New Business • Tax	Yes
Charge Type Sub Category	Used to indicate the subcategory to which the charge type belongs.	No

Tip: Alternatively, you can access this screen by clicking the **Edit** button in the **Charge Type** zone.

4. Modify the required details in the **Charge Type** screen.
5. Define, edit, or remove characteristics of the charge type, if required.
6. Click **Save**.
   The changes made to the charge type are saved.

Related Topics

For more information on...	See...
Charge Type screen	Charge Type on page 246
Charge Type List zone	Charge Type List on page 246
Charge Type zone	Charge Type on page 247
How to define a characteristic for a charge type	Defining a Characteristic for a Charge Type on page 249

Copying a Charge Type

Instead of creating a charge type from scratch, you can create a new charge type using an existing charge type. This is possible through copying a charge type. On copying a charge type, the details including the characteristics are copied to the new charge type. You can then edit the details, if required.

Prerequisites

To copy a charge type, you should have:

- Values defined for the **CHG_TYPE_SUB_CAT** lookup field

Procedure

To copy a charge type:

1. Click the **Admin** link in the **Application** toolbar.

Copyright © 2024, Oracle and/or its affiliates.

A list appears.

2. From the **Admin** menu, select **C** and then click **Charge Type**.
   A sub-menu appears.
3. Click the **Search** option from the **Charge Type** sub-menu.
   The **Charge Type** screen appears.
4. In the **Charge Type List** zone, click the **Duplicate** (⧉) icon in the **Duplicate** column *c*orresponding to the charge type whose copy you want to create.

   The **Charge Type** screen appears. It contains the following sections:

   - **Main** - Used to specify basic details about the charge type.
   - **Characteristics** - Used to define the characteristics for the charge type.

   The **Main** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Charge Type	Used to specify the charge type.	Yes
Description	Used to specify the description for the charge type.	Yes
Charge Type Category	Used to indicate the category to which the charge type belongs. The valid values are: • Additional taxes • Commission • Down Payment • Fees • Premium- Endorsement • Premium- New Business • Tax	Yes
Charge Type Sub Category	Used to indicate the subcategory to which the charge type category belongs.	No

Tip: Alternatively, you can copy a charge type by clicking the **Duplicate** button in the **Charge Type** zone.

5. Enter the required details in the **Main** section.
6. Define, edit, or remove characteristics of the charge type, if required.
7. Click **Save**.
   The new charge type is defined.

Related Topics

For more information on...	See...
Charge Type screen	Charge Type on page 246
Charge Type List zone	Charge Type List on page 246
Charge Type zone	Charge Type on page 247
How to define a characteristic for a charge type	Defining a Characteristic for a Charge Type on page 249

Deleting a Charge Type

Procedure

Copyright © 2024, Oracle and/or its affiliates.

To delete a charge type:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **C** and then click **Charge Type**.
   A sub-menu appears.
3. Click the **Search** option from the **Charge Type** sub-menu.
   The **Charge Type** screen appears.
4. In the **Charge Type List** zone, click the **Delete** (🗑) icon in the **Delete** column corresponding to the charge type that you want *t*o delete.

   A message appears confirming whether you want to delete the charge type.

   Note: You can delete a charge type which is not yet linked to any contract type.

   Tip: Alternatively, you can delete a charge type by clicking the **Delete** button in the **Charge Type** zone.
5. Click **OK**.
   The charge type is deleted.

Related Topics

For more information on...	See...
Charge Type screen	Charge Type on page 246
Charge Type List zone	Charge Type List on page 246

Viewing the Charge Type Details

Procedure

To view the details of a charge type:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **C** and then click **Charge Type**.
   A sub-menu appears.
3. Click the **Search** option from the **Charge Type** sub-menu.
   The **Charge Type** screen appears.
4. In the **Charge Type List** zone, click the **Broadcast** (📶) icon corresponding to the charge type whose details you want to view.
   The **Charge Type** zone appears.
5. View the details of the charge type in the **Charge Type** zone.

Related Topics

For more information on...	See...
Charge Type screen	Charge Type on page 246
Charge Type List zone	Charge Type List on page 246
Charge Type zone	Charge Type on page 247

Copyright © 2024, Oracle and/or its affiliates.

Maintaining Billable Charges

You create a billable charge whenever a customer should be charged for a service that occurs outside the normal course of business.

Note: Billable charges can be uploaded from an external system. Refer to Uploading Billable Charges and Using Billable Charges For Pass Through Billing for more information.

Note: If the billable charges are generated automatically through the transaction feed management or on demand billing process, you cannot edit the details of the billable charge. You can only cancel a billable charge, if required.

A billable charge must refer to a contract. This contract behaves just like any other contract:

- **Bill segments are created for the contract.** Whenever billing is performed for an account with billable charge contracts, the system creates a bill segment for each billable charge. This means that if a contract has many unbilled billable charges, it will have many bill segments.
- **Payments are distributed to the contract.** Payments made by an account are distributed to its billable charge contracts just like any other contract.
- **Overdue debt is monitored.** The credit and collections process monitors billable charge contracts for overdue debt and responds accordingly when overdue debt is detected.
- This contract must reference an contract type that has a Special Role Flag of `Billable Charge`. If such a contract does not already exist for the account, one must be created before you can levy a billable charge. See "For more information" at the end of this section for links to other sections that will help you with this process.

Billable charge *templates* exist to minimize the effort required to create a billable charge for a customer. A billable charge template contains the standard bill lines, amounts and distribution codes used to charge for a one-off charge.

A user may override the information on the template when the billable charge is created. This template would contain the bill lines, amounts and distribution codes associated with a ruptured pipe repair. Then, when a contractor ruptures a line, a user can create a billable charge using the template and override the amount to reflect the actual repair amount (if it differs from the norm).

Note: Templates aren't required. A billable charge can be created without a template for a truly unexpected charge. If you don't use a template, you'll have to enter the bill information manually.

Fastpath: Refer to Setting Up Billable Charge Templates for more information.

Note: Adding taxes and other calculation lines to a billable charge. Refer to Billable Charge - SQ Details for information about how to have the system calculate tax lines and append them to the billable charge.

Billable Charge - Main

Open **Billing Management, Billable Charge** to add or update a billable charge.

Description of Page

The **Billable Charge ID** is system generated. The **Account ID** of the account responsible for the charge is displayed at the top.

Use **Contract ID** to define the ID of the billable charge contract that will hold the billable charge's debt. The **Account ID** for the contract is displayed for information purposes.

Use **Start Date** and **End Date** to define the charge's bill period.

Copyright © 2024, Oracle and/or its affiliates.

> **Note: Installment billable charges.** You can create many future dated billable charges to implement unusual installment plans. For example, if you want to bill a customer $250 today, $150 next month, and $1000 three months from now; you could create three billable charges with a date that reflects the earliest bill date. When a bill is produced for the account, the system will create a bill segment for every billable charge whose start date is on or before the current date.

If you want to create the billable charges from an existing template, enter the ID of the **Billable Charge Template** from which the billable charge information defaults. If your contract type indicates a billable charge template, that value will be defaulted.

Use **Description on Bill** to define the verbiage that should print on the customer's bill above the line item details.

Use the **Charge Type** field to indicate the type of billable charge you want to create.

Billable Charge Status defines the state of the billable charges. Possible values are `Billable` and `Cancelled`. Click the **Cancel** button to cancel a billable charge.

Use the **Recurring Flag** field to indicate whether recurring bill segments must be created at the intervals defined in the bill period or at the set invoice frequency. The valid values are:

- Bill Period
- Frequency

Use the **Frequency** field to indicate the invoice frequency at which the recurring bill segments must be created. This field is enabled when you select the **Frequency** option from the **Recurring Flag** list.

> **Note:** The invoice frequency is used in combination with the contract start date to arrive at the date from when the recurring bill segments must be created. For example, if the invoice frequency is monthly and the contract start date is 4th October, the recurring bill segments will be created on 4th of every month from October onwards.

Use the **Bill Period** field to indicate the bill period that you want to use for specifying the intervals at which the recurring bill segments must be created. This field is enabled when you select the **Bill Period** option from the **Recurring Flag** list.

Use the **Bill After** field to specify the date after which the bill must be generated for the billable charge.

Use the **Adhoc Bill** check box to indicate whether the billable charge should be considered during adhoc or regular billing.

Bill Segment ID is the ID of the bill segment on which the billable charge's charge details appear. This information is only populated AFTER the billable charge has been swept onto a bill.

The **Total Bill Amount** displays the sum of all billable charge lines except those that are **Memo Only**. The **Total Line Amount** displays the sum of all billable charge lines except those that **Appear In Summary**.

The information in the scroll defines the line item details associated with the billable charge. Click the drill down button to navigate to the line characteristics. The following fields are required for each line:

Sequence Sequence controls the order in which the line items appear on the bill segment.

Description on Bill Specify the verbiage to print on the bill for the line item.

Currency Specify the currency in which the amount is charged. Note that if you specify a currency other than the account's invoice currency, the system will do the currency conversion if the appropriate exchange rate is available in the system.

Charge Amount Specify the amount to charge for the line item.

Show on Bill Turn this switch on if the line item should appear on the customer's printed bill. It would be very unusual for this switch to be off.

Appears in Summary Turn this switch on when the amount associated with this line also appears in a summary line. This switch plays a part at bill print time - those lines that appear in a summary print in the left dollar column, those that don't appear in a summary print in the right dollar column. Also note, those lines that appear in a summary line are not included in the total amount associated with the billable charge.

Copyright © 2024, Oracle and/or its affiliates.

Memo Only, No GL Turn this switch on if the billable charge line does not contribute to the total amount due (e.g., this switch would be on for "information only" lines).

Distribution Code Specify the distribution code that identifies the GL account associated with this line item. The description of the code appears adjacent.

Note: If a billable charge is generated automatically through the transaction feed management or on demand billing process, you cannot edit the details of the billable charge. You can only cancel the billable charge, if required.

Fastpath: For more information about billing billable charges, refer to How To Create A One-Time Invoice. For more information about billable charge templates, refer to Setting Up Billable Charge Templates.

Billable Charge - Line Characteristics

Characteristics may be associated with billable charge lines. Typically, this is done to categorize billable charge lines for reporting purposes.

Open **Billing Management, Billable Charge, Line Characteristics** to maintain characteristics that are associated with billable charge.

Description of Page

The **Billable Charge Line** scroll defines the billable charge line to which you wish to assign characteristic values. The following fields display:

Characteristic Type The type of characteristic.

Characteristics Value The value of the characteristic.

Note: You can only choose characteristic types defined as permissible on a billable charge line. Refer to Setting Up Characteristic Types & Their Values for more information.

Note: If the billable charges are generated automatically through the transaction feed management or on demand billing process, you cannot edit the details of the billable charge. You can only cancel a billable charge, if required.

Billable Charge - SQ Details

Optionally, you may specify service quantities associated with a billable charge. These are most commonly used when you want the system to calculate charges *in addition to the billable charge lines* (e.g., adding tax to the billable charge lines).

Note: The contract must specify a rate. If you want the system to calculate additional charges based on the service quantities defined on this page, the related billable charge contract must reference a rate. This rate must have rate components that calculate charges based on the service quantities specified on this page. When the contract is next billed, the system will sweep on the billable charges and then call the rate and pass it the service quantities defined on this page. The rate will calculate the additional charges and these will be appended to the bill segment.

Open **Billing Management, Billable Charge, SQ Details** to maintain a billable charges service quantities.

Description of Page

The following fields display:

Sequence Number Used to indicate the sequence number of the service quantity.

UOM Used to indicate the product for which you want to create the billable charge.

Variance Parameter Used to indicate the variance parameter which you want to use along with the product for determining the price.

Copyright © 2024, Oracle and/or its affiliates.

Service Qty. Identifier Used to indicate the service quantity identifier based on which you want to bill the customer.

Service Quantity Used to specify the total service quantity for which you want to create the billable charge.

Note: If the billable charges are generated automatically through the transaction feed management or on demand billing process, you cannot edit the details of the billable charge. You can only cancel a billable charge, if required.

Billable Charge - Characteristics

The **Characteristics** tab allows you to define characteristics for a billable charge. This tab contains the following fields:

Field Name	Field Description
Effective Date	Used to specify the date from when the characteristic is effective for the billable charge.
Characteristic Type	Used to specify the characteristic type.
Characteristic Value	Used to specify the value for the characteristic type.
	Note: When you specify the value for a predefined characteristic type, the description of the characteristic value appears corresponding to the **Characteristic Value** field. When you select a predefined characteristic type, the **Search** icon appears corresponding to the **Characteristic Value** field. On clicking the **Search** icon, you can search for a predefined characteristic value.

Note: If the billable charges are generated automatically through the transaction feed management or on demand billing process, you cannot edit the details of the billable charge. You can only cancel a billable charge, if required.

Billable Charge - Pricing Information

If you are manually creating a billable charge, the **Pricing Information** tab allows you to specify the details of the product pricing which is applicable to the billable charge. However, if a billable charge is automatically generated (through the Transaction Feed Management or On Demand Billing process), the **Pricing Information** tab allows you to view the details of the product pricing which is applicable to the billable charge. This tab contains the following fields:

Field Name	Field Description
Product	Used to indicate the product for which the billable charge is created.
Variance Parameter	Used to indicate the variance that must be used along with the product for determining the product pricing.
	Note: This field is enabled when the **Multi Price Parameter** option type of the **Multi Parameter Based Pricing (C1_PPARM_FLG)** feature configuration is set to **False**.
Price Assignment ID	Indicates the product pricing which is applicable to the billable charge.
	Note: The data in this field appears when the billable charge is automatically generated through the Transaction Feed Management process.
Person ID	Indicates the customer whose product pricing is applicable to the billable charge.

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description
	Note: The data in this field appears when the billable charge is automatically generated through the Transaction Feed Management process.
Account ID	Indicates the account whose product pricing is applicable to the billable charge.
	Note: The data in this field appears when the billable charge is automatically generated through the Transaction Feed Management process.
Price List ID	Indicates the price list whose product pricing is applicable to the billable charge.
	Note: The data in this field appears when the billable charge is automatically generated through the Transaction Feed Management process.
Product Parameter	Used to indicate the product parameter that must be used along with the product for determining the product pricing.
	Note: This field is enabled when the **Multi Price Parameter** option type of the **Multi Parameter Based Pricing (C1_PPARM_FLG)** feature configuration is set to **True**.
Product Parameter Value	Used to specify the value of the parameter.
	Note: You can specify any special characters except ampersand (&), comma (,), underscore (_), equal to (=), tilde (~), and semicolon(;) in the parameter value. This field is enabled when the **Multi Price Parameter** option type of the **Multi Parameter Based Pricing (C1_PPARM_FLG)** feature configuration is set to **True**.

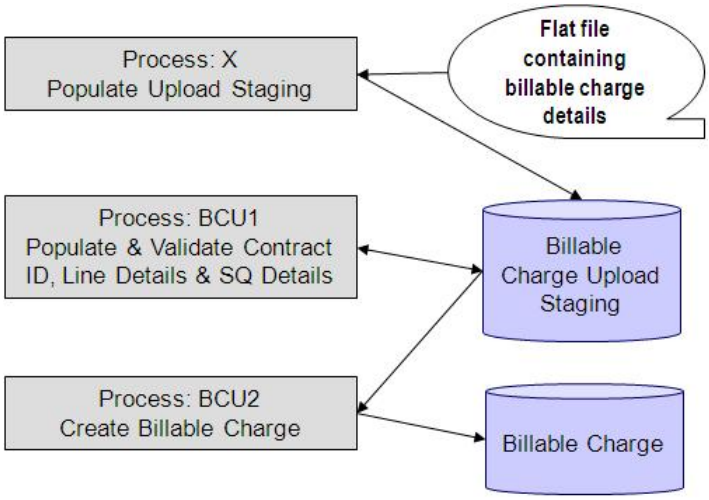
Uploading Billable Charges

This section describes how the system uploads billable charges from an external source. This mechanism is used when you need to "pass through" charges that have been calculated by a third party but appear on your bill.

Billable Charge Upload Background Processes

The following diagram illustrates the processes involved in the uploading of billable charges into the system.

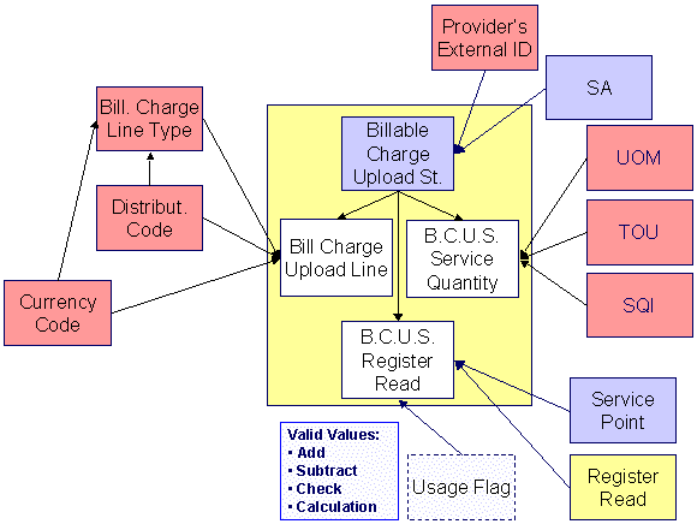
Copyright © 2024, Oracle and/or its affiliates.



The topics in this section describe how these processes work.

Process X - Populate BC Upload Staging

Process X refers to the mechanism used by your organization to populate the various staging tables (shown in the yellow section of the following ERD).



The topics in this section describe each of these tables.

Billable Charge Upload Staging Layout

You must create an upload staging record for each billable charge. The name of this table is CI_BCHG_UP . The following table describes each column on this table.

Column Name	Length	Required	Data Type	Comments
BCHG_UP_ID	30	Yes	CHAR	This is the unique identifier of the record. This value does NOT have to be a random number, but it does need to be unique. If your process that inserts records on this table is capable of calling a COBOL routine, call CIPCKEYG and it will supply a 12 digit random number for you.
BCHG_UP_STAT_FLG	2	Yes	CHAR	This must be set to `P` for `Pending`.

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Length	Required	Data Type	Comments
NT_XID_CD	30	No	CHAR	This is the external system id of the sender of the billable charge. Refer to Designing External Systems for more information about this field.
BCHG_EXT_REF_ID	50	No	VARCHAR2	This is the identifier of the billable charge in the sender's system.
CRE_DTTM	26	No	DATE/TIME	The date and time the upload staging row was inserted (this can be used for audit purposes).
START_DT	10	Yes	DATE	The start date of the period encompassed by the billable charge.
END_DT	10	Yes	DATE	The end date of the billable charge period.
DESCR_ON_BILL	80	Yes	VARCHAR2	This is the description that should prefix the charges on the printed bill.
SA_ID	10	See note	CHAR	This must correspond with a contract ID of a billable charge contract.
BILLABLE_CHG_ID	12	No	CHAR	Leave this column blank. It will be assigned by the system when it creates a billable charge record.
CHG_TYPE_CD	12	No	CHAR	This column is used to specify the charge type for which you want to create a billable charge.
RECURRING_FLG	4	No	CHAR	This column is used to indicate whether recurring bill segments must be created at the intervals defined in the bill period or at the set invoice frequency. The valid values are: • BP • FR
POLICY_INVOICE_FREQ_CD	4	No	CHAR	This column is used to indicate the invoice frequency at which the recurring bill segments must be created.
BILL_PERIOD_CD	8	No	CHAR	This column is used to indicate the bill period that you want to use for specifying the intervals at which the recurring bill segments must be created.

Billable Charge Line Upload Staging Layout

You must create a billable charge line upload record for each line to be uploaded. The name of this table is CI_BCHG_LINE_UP . The following table describes each column on this table.

Note: Defaulting from BCHG_UP_XTYPE. Please pay special attention to the **Comments** column below as many fields, if left blank, will default from each record's BCHG_UP_XTYPE.

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Length	Req'd	Data Type	Comments
BCHG_UP_ID	30	Y	Char	This is the foreign key to the billable charge upload staging record.
LINE_SEQ	3	Y	N	This is the unique line number of the billable charge line. This value must be unique within the billable charge.
DESCR_ON_BILL	80	Y	Varchar2	This is the description that will be printed on the bill for the line.
CHARGE_AMT	13.2	N	N	This is the amount associated with the billable charge line.
BCHG_LINE_XID	20	N	Varchar2	This is the unique identifier of the billable charge line in the sender's system.
BCHG_UP_XTYPE	30	N	Char	This is the type of billable charge line. This is a foreign key reference to the billable charge line type control table. This control table is used to populate the remaining fields (if they are blank). If this field is blank, the remaining fields must be specified as they cannot be defaulted from the billable charge line type control table. Refer to *Billable Charge Line Type* for more information.
CURRENCY_CD	3	See note	Char	This is a foreign key reference to the currency code of the billable charge line. This field must be specified if CHARGE_AMT is non-zero.
SHOW_ON_BILL_SW	1	See note	Char	This should be `Y` if the line should appear on the customer's printed bill. This should be `N` if the line should not appear on the customer's printed bill. If left blank, this value will be populated from the line's BCHG_UP_XTYPE.
APP_IN_SUMM_SW	1	See note	Char	This should be `Y` if the line appears in a summary line. This should be `N` if the line does not appear in a summary line. This switch plays a part at bill print time - those lines that appear in a summary print in the left dollar column, those that don't appear in a summary print in the right dollar column. If left blank, this value will be populated from the line's BCHG_UP_XTYPE.
MEMO_SW	1	See note	Char	This should be `Y` if the line does NOT affect the general ledger. This should be `N` if the line affects the general ledger. If left blank, this value

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Length	Req'd	Data Type	Comments
				will be populated from the line's BCHG_UP_XTYPE.
DST_ID	10	See note	Char	If the line affects the general ledger, the set ID (the previous field) and the distribution code of the GL account must be specified. If left blank, this value will be populated from the line's BCHG_UP_XTYPE.

Billable Charge Line Characteristic Upload Staging Layout

If you want to upload characteristics on your billable charge lines, you must create a billable charge line characteristic upload record for each characteristic on each line. The name of this table is CI_B_LN_UP_CHAR . The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
BCHG_UP_ID	30	Y	Char	This is the foreign key to the billable charge upload staging record.
LINE_SEQ	3	Y	N	This is the foreign key to the billable charge line upload staging record
CHAR_TYPE_CD	8	Y	Char	This is the unique identifier of the billable charge line characteristics. This value must be unique within the billable charge line characteristics.
CHAR_VAL	16	N	Char	A characteristic value must be supplied if the ad hoc switch on characteristic type is `N`. Otherwise, it is not allowed.
ADHOC_CHAR_VAL	30	N	Varchar2	An ad hoc characteristic value must be supplied if the ad hoc switch on characteristic type is `Y`. Otherwise, it is not allowed.

Billable Charge Service Quantity Upload Staging Layout

You must create a billable charge service quantity upload record for each service quantity to be uploaded. The name of this table is CI_BCHG_UP_SQ . The following table describes each column on this table. Refer to "pass through" charges for more information about how this type of information is used.

Column Name	Length	Req'd	Data Type	Comments
BCHG_UP_ID	30	Y	Char	This is the foreign key to the billable charge upload staging record.
SEQ_NUM	3	Y	N	Sequence of the service quantity within the Billable Charge Upload
UOM_CD	4	N	Char	The unit of measure of the Billable Charge Upload SQ row. If specified, it must

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Length	Req'd	Data Type	Comments
				reference a valid value on CI_UOM. Note, at least one of the following fields should be specified: SQI_CD, TOU_CD, UOM_CD.
TOU_CD	8	N	Char	The Variance Parameter of the Billable Charge Upload SQ row. If specified, it must reference a valid value on CI_TOU. Note, at least one of the following fields should be specified: SQI_CD, TOU_CD, UOM_CD.
SQI_CD	8	N	Char	The service quantity identifier of the Billable Charge Upload SQ row. If specified, it must reference a valid value on CI_SQI. Note, at least one of the following fields should be specified: SQI_CD, TOU_CD, UOM_CD.
SVC_QTY	12.6	N	N	Quantity of the service used for the UOM / Variance Parameter/ SQI combination

Billable Charge Characteristic Upload Staging Layout

If you want to upload characteristics on your billable charge, you must create a billable charge characteristic upload record for each characteristic. The name of this table is CI_BCHG_UP_CHAR . The following table describes each column on this table:

Column Name	Length	Required	Data Type	Comments
ADHOC_CHAR_VAL	254	No	VARCHAR2	This column is used to specify an adhoc characteristic value. The data is required if the char value type of the characteristic type is set to **Adhoc Value**.
BCHG_UP_ID	30	Yes	CHAR	This column is used to specify unique ID for billable charge upload staging record.

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Length	Required	Data Type	Comments
CHAR_TYPE_CD	8	Yes	CHAR	This column is used to specify the characteristic type. The data specified in this column must exists in the system.
CHAR_VAL	16	No	CHAR	This column is used to specify the characteristic value. The data is required if the char value type of the characteristic type is set to **File Location Value** or **Predefined Value**.
CHAR_VAL_FK1	50	No	VARCHAR2	This column is used to specify a foreign key. The data is required if the char value type of the characteristic type is set to **Foreign Key Value**.
CHAR_VAL_FK2	50	No	VARCHAR2	This column is used to specify a foreign key. This column can be used when you have more than one foreign key.
CHAR_VAL_FK3	50	No	VARCHAR2	This column is used to specify a foreign key. This column can be used when you have more than one foreign key.
CHAR_VAL_FK4	50	No	VARCHAR2	This column is used to specify a foreign key. This column can be used when you have more than one foreign key.
CHAR_VAL_FK5	50	No	VARCHAR2	This column is used to specify a foreign key. This column can be used when you have more than one foreign key.
EFFDT	10	Yes	DATE	This column is used to specify the date from when the characteristic is effective for the billable charge upload record.
SRCH_CHAR_VAL	254	No	VARCHAR2	This column is used to specify the description of the predefined characteristic value.
VERSION	5	Yes	NUMBER	This column is used to specify the version number.

BCU1 - Validate & Populate Billable Charge Upload Staging

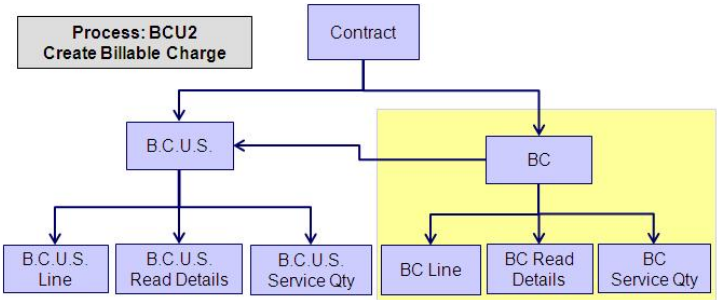
This process populates various fields (e.g., the GL distribution code, memo only switch) on the billable charge upload line records from the billable charge line type specified on each respective record.

Any validation / population errors detected during this process are written to the BC Upload Exception table. You can fix errors using BC Upload Staging page (don't forget to change the record's status back to `Pending`).

BCU2 - Create Billable Charge

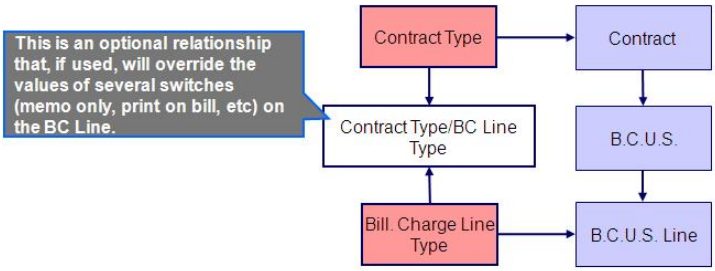
This process creates a billable charge and billable charge lines (shown in the yellow section of the following ERD) for all BC upload records in the `Pending` state.

Copyright © 2024, Oracle and/or its affiliates.



If the billable charge's contract is `Closed`, it will `Reactivate` it so that the billable charges will be swept onto the next bill produced for the account.

Note: This process will override the values of the various switches referenced on bill charge upload staging line if the respective contract's contract type has an override value for the line's billable charge line type. Refer to Contract Type - Billable Charge Overrides for more information.



Any errors detected during this process are written to the Billable Charge Upload Exception table. You can fix errors using BC Upload Staging page (don't forget to change the record's status back to `Pending`).

BCUP-PRG - Purge Billable Charge Upload Objects

`Completed` billable charge upload staging objects should be periodically purged from the system by executing the BCUP-PRG background process. This background process allows you to purge all `Completed` billable charge upload staging objects older than a given number of days.

We want to stress that there is no system constraint as to the number of `Completed` billable charge upload objects that may exist. You can retain these objects for as long as you desire. However we recommend that you periodically purge `Completed` billable charge upload objects as they exist only to satisfy auditing and reporting needs.

Billable Charge Upload Staging

The Billable charge Upload Staging page has three purposes:

- You can view historical billable charge upload records.
- You can correct billable charge upload records that are in error.
- You can add new billable charge upload records. These will be uploaded by the billable charge upload process (although it's probably easier to just add the billable charges using the Maintaining Billable Charges page).

The topics in the section describe how to maintain this information.

Billable Charge Upload - Main

Open this page using **Accounting and Receivable, Billable Charge Upload Staging, Main**.

Description of Page

Copyright © 2024, Oracle and/or its affiliates.

Fastpath: Refer to Billable Charge Upload Staging for more information about these fields.

Note: Correcting Errors. If a Billable Charge Upload record's status is `Error`, you should correct the problem and then change its status back to `Pending`. The next time the billable charge upload process runs, it will revalidate the record and, if valid, it will create a billable charge record.

Billable Charge Upload - Lines

Open this page using **Accounting and Receivable, Billable Charge Upload Staging, Lines**.

Description of Page

Fastpath: Refer to Billable Charge Line Upload Staging for more information about these fields.

Billable Charge Upload - SQ Details

Open this page using **Accounting and Receivable, Billable Charge Upload Staging, SQ Details**.

Description of Page

Fastpath: Refer to Billable Charge Service Quantity Upload Staging for more information about these fields.

Billable Charge Upload - Characteristics

Open this page using **Accounting and Receivable, Billable Charge Upload Staging, Characteristics**.

Description of Page

Fastpath: Refer to Billable Charge Characteristic Upload Staging Layout on page 263 for more information about these fields.

Billable Charge Upload Exception

If errors are detected during the billable charge upload process, a record is written to the billable charge upload exception table with a message indicating the nature of the severe error.

To view the messages associated with the exception records, schedule the TD-BCUPL background process. This process generates a To Do entry for every record in the billable charge upload exception table.

You can fix this error using the Billable Charge Upload Staging page and change the status of the record from `Error` to `Pending`. When the billable charge upload process next runs, it attempts to upload this record again.

Budget Review

The Budget Review page shows all accounts with an anomalous budget amount.

Note: This information on this page is created by the Budget Monitor background process.

Open this page using **Report and Inquiry, Budget Review**.

Description of Page

One row is displayed for every account whose current budget amount is out-of-sync with the recommended budget amount.

Copyright © 2024, Oracle and/or its affiliates.

Note: Rows are also displayed for errors. Please be aware that if the Budget Monitor background process encounters an error when it attempts to calculate an account's recommended budget, these errors will also appear as rows in this query.

Bill Segment Summary

The **Bill Segment Summary** screen allows you to search for bill segments using various search criteria. This screen consists of the following zones:

- Search on page 267

Search

The **Search** zone allows you to search for bill segments using various search criteria. The fields in the **Search Criteria** section change depending on the query option selected. You can select either of the following query options from the **Search Using** list:

- **Account ID** - Used to indicate that you want to search bill segments of a particular account.
- **Bill Segment ID** - Used to indicate that you want to search for a particular bill segment.
- **Bill Details** - Used to indicate that you want to search bill segments with the specific billing details, such as bill segment type, bill segment status, and so on.
- **Contract ID** - Used to indicate that you want to search bill segments created for a particular contract.
- **Person Details** - Used to indicate that you want to search bill segments created for a particular person or price item, or created with particular billing method or invoice currency, or created in particular bill cycle.
- **Person Name** - Used to indicate that you want to search bill segments created for a particular person.

The **Search** zone contains the following two sections:

- **Search Criteria** - The **Search Criteria** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Account ID	Used to indicate the invoice account whose bill segments you want to search. You can search for an account by clicking the **Search** (🔍) icon corresponding to the field. **Note:** This field appears when you select the **Account ID** option from the **Search Using** list.	Yes (Conditional) **Note:** This field is required when you select the **Account ID** option from the **Search Using** list.
Bill Segment ID	Used to indicate the bill segment you want to search. **Note:** This field appears when you select the **Bill Segment ID** option from the **Search Using** list.	Yes (Conditional) **Note:** This field is required when you select the **Bill Segment ID** option from the **Search Using** list.
Bill ID	Used to indicate the bill whose bill segments you want to search. **Note:** This field appears when you select the **Bill Details** option from the **Search Using** list.	Yes (Conditional) **Note:** This field is required when you select the **Bill Details** option from the **Search Using** list.

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
Bill Segment Status	Used to search for bill segments with a particular status. The valid values are: • Canceled • Error • Freezable • Frozen • Incomplete • OK • Pending Cancel **Note:** This field appears when you select the **Bill Details** option from the **Search Using** list.	No
Start Date	Used to search for bill segments with a particular start date. **Note:** This field appears when you select the **Bill Details** option from the **Search Using** list.	No
End Date	Used to search for bill segments with a particular end date. **Note:** This field appears when you select the **Bill Details** option from the **Search Using** list.	No
Bill Segment Type	Used to search for bill segments with a particular bill segment type. The valid values are: • Post Processing • Regular **Note:** This field appears when you select the **Bill Details** option from the **Search Using** list.	No
Usage Account ID	Used to indicate the usage account whose bill segments you want to search. **Note:** This field appears when you select the **Bill Details** option from the **Search Using** list.	No
Contract ID	Used to search for bill segments which are created for a particular contract. **Note:** This field appears when you select the **Contract ID** option from the **Search Using** list.	Yes (Conditional) **Note:** This field is required when you select the **Contract ID** option from the **Search Using** list.
Person ID	Used to indicate the person whose bill segments you want to search. You can search for a person by clicking the **Search** (🔍) icon corresponding to the field.	Yes (Conditional) **Note:** This field is required when you select the **Person Details** option from the **Search Using** list.

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: This field appears when you select the **Person Details** option from the **Search Using** list.	
Bill Segments of Persons in Hierarchy	Used to indicate whether you want to search bill segments of all persons in the hierarchy. **Note:** This field appears when you select the **Person Details** option from the **Search Using** list.	No
Bill Cycle	Used to search for bill segments which are created in a particular bill cycle. **Note:** This field appears when you select the **Person Details** option from the **Search Using** list.	No
Invoice Currency	Used to search for bill segments which are created with a particular invoice currency. **Note:** This field appears when you select the **Person Details** option from the **Search Using** list.	No
Billing Method	Used to search for bill segments which are created during adhoc or regular billing. The valid values are: • Adhoc Billing • Regular Billing **Note:** This field appears when you select the **Person Details** option from the **Search Using** list.	No
Price Item	Used to search for bill segments which are created for a particular price item. **Note:** This field appears when you select the **Person Details** option from the **Search Using** list.	No
Person Name	Used to indicate the person whose bill segments you want to search. **Note:** This field appears when you select the **Person Name** option from the **Search Using** list.	Yes (Conditional) **Note:** This field is required when you select the **Person Name** option from the **Search Using** list.

- **Search Results** - On clicking the **Search** button, the search results appear based on the specified search criteria. The **Search Results** section contains the following columns:

Column Name	Column Description
Bill Segment ID	Displays the bill segment ID.
Bill Segment Information	Displays additional information about the bill segment. In addition, this column has a context menu which helps in navigating to other screens in the application. **Note:** It has a link. On clicking the link, the **Bill Segment** screen appears with the details of the respective bill segment.

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
Start Date	Displays the start date of the bill segment.
End Date	Displays the end date of the bill segment.
Bill Segment Status	Indicates the current status of the bill segment. The valid values are: • Canceled • Error • Freezable • Frozen • Incomplete • OK • Pending Cancel
Amount	Displays the amount of the bill segment.
Price Item	Indicates the price item for which the bill segment is created.
Invoice Account	Indicates the invoice account for which the bill segment is created.
Usage Account ID	Indicates the usage account for which the bill segment is created.
Bill ID	Indicates the bill to which the bill segment belongs.
Bill Information	Displays additional information about the bill. In addition, this column has a context menu which helps in navigating to other screens in the application.
	Note: It has a link. On clicking the link, the **Bill** screen appears with the details of the respective bill.
Person Name	Indicates the person for whom the bill segment is created.

Related Topics

For more information on...	See...
How to search for a bill segment	Searching for a Bill Segment on page 270

Searching for a Bill Segment

Prerequisites

To search for a bill segment, you should have:

- Bill cycles and currencies defined in the application

Procedure

To search for a bill segment:

1. Click the **Menu** link in the **Application** toolbar.
   A list appears.
2. From the **Main** menu, select **Billing Management** and then click **Bill Segment Summary**.
   The **Bill Segment Summary** screen appears.
3. Select the query option, using which you want to search for a bill segment, from the **Search Using** list.
   The fields appear in the **Search Criteria** section depending on the query option selected.
4. Enter the search criteria in the **Search** zone.

Copyright © 2024, Oracle and/or its affiliates.

> **Note:** ORMB search engine supports wildcard search, where you can substitute the percentage (%) symbol as a stand in for any word or letter in a search criteria. You can use the '%' wildcard character in all input fields except the date and ID fields. The '%' wildcard character is suffixed automatically at the end of the partial search criteria. Therefore, you may or may not specify the wildcard character at the end of the partial search criteria. However, you have to prefix the wildcard character manually wherever required.

5. Click **Search**.
   A list of bill segments that meet the search criteria appears in the **Search Results** section.
6. If required, you can export the list of bill segments in the Excel format by clicking the **Export To Excel** menu option from the **Explorer Zone** (▤) menu.

Related Topics

For more information on...	See...
Bill Segment Summary screen	Bill Segment Summary on page 267
Search zone	Search on page 267

Bill Drill Down Details

The **Bill Drill Down Details** screen enables you to trace the source of various billing elements, such as bill and billable charge. It contains the following zones:

- **Search Billing Element** - Enables you to search for a bill or a billable charge using various search criteria.
- **Bill Segments** - Enables you to view the bill segments of the bill.
- **SQ Details** - Enables you to view the service quantity identifiers (SQIs) used in the bill segment calculation. It appears when the bill segment is generated for an SQI based billable charge.
- **Bill Segment Calculation Lines** - Enables you to view the calculation lines of the bill segment.
- **Timeline and Billable Charge Details** - Enables you to view the timelines which are derived during the membership premium calculation. It appears when the timeline details are available for the billable charge for which the bill segment is created.
- **Pricing Calculation Details** - Enables you to view how the amount is calculated for the timeline. If the age based pricing is used during the membership premium calculation, this zone lists the premium calculated for each eligible member of the membership. It also displays the base fee and modifier fee which are applicable for each eligible member. However, if the tier based pricing is used during the membership premium calculation, this zone lists the premium calculated at the membership level.
- **Transaction Details** - Enables you to view the transactions whose charges are billed through the bill segment. It appears when the bill segment is generated for a billable charge which is created through the transaction aggregation process.
- **Transaction Leg Calculation Lines** - Enables you to view calculation lines of all legs of the transaction which are rated before billing.

Search Billing Element

The **Search Billing Element** - zone allows you to search for a bill using various search criteria. This zone contains the following two sections:

- **Search By** - The **Search Criteria** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Bill ID	Used to search a particular bill.	No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: The **Search** (🔍) icon appears corresponding to the **Bill ID** field. On clicking the **Search** icon, the **Bill Search** window appears.	
Division	Used to search bills of accounts that belong to a particular division. **Note:** You can only view those divisions to which you have access in the **Division** list. One more search criterion is required when you are searching for a bill using the **Division** field.	Yes (Conditional) **Note:** This field is not required when you search for a bill using the bill ID or alternate bill ID.
Bill Status	Used to search bills with a particular status. The valid values are: • All • Complete • Pending	No
Alternate Bill ID	Used to search a particular bill using the alternate bill ID.	No
Bill Completion From Date	Used to search bills which are completed on or later than the specified date.	No
Bill Completion To Date	Used to search bills which are completed on or earlier than the specified date.	No
Person ID	Used to search bills of a particular person. **Note:** The **Search** (🔍) icon appears corresponding to the **Person ID** field. On clicking the **Search** icon, the **Person Search** window appears.	No
Primary Account Identifier Type	Used to select the identifier type based on which you want to search bills of a particular account.	Yes (Conditional) **Note:** If you enter the account identifier as a search criteria, you have to select the account identifier type.
Primary Person Identifier Type	Used to select the identifier type based on which you want to search bills of a particular person.	Yes (Conditional) **Note:** If you enter the person identifier as a search criteria, you have to select the person identifier type.
Primary Account Identifier	Used when you want search bills of a particular account.	Yes (Conditional)

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
		Note: If you specify the account identifier type as a search criteria, you have to enter the account identifier.
Primary Person Identifier	Used when you want search bills of a particular person.	Yes (Conditional) **Note:** If you specify the person identifier type as a search criteria, you have to enter the person identifier.
Billable Charge ID	Used to search a particular billable charge information using the billable charge ID. **Note:** The **Search** (🔍) icon appears corresponding to the **Billable Charge ID** field. On clicking the **Search** icon, the **Billable Charge Search** window appears. This field appears when you select **Billable Charge Details** option from the **Search By** list.	Yes

- **Search Results** - On clicking the **Search** button, the search results appear based on the specified search criteria. The **Search Results** section contains the following columns:

Column Name	Column Description
Bill ID	Displays the bill ID. **Note:** It has a link. On clicking the link, the **Bill** screen appears with the details of the respective bill.
Status	Displays the status of the bill.
Completion Date	Displays the date when the bill was completed.
Total Bill Amount	Displays the total amount billed to the person.
Invoice Currency	Displays the currency in which the amount is billed to the person.
Account Identifier Type	Displays the primary account identifier type.
Account Identifier	Indicates the account for which the bill is created.
Person Identifier Type	Displays the primary person identifier type.
Person Identifier	Indicates the person for whom the bill is created.
Division	Displays the division to which the account belongs.
Alternate Bill ID	Displays the alternate bill ID.
Start Date	Displays the date from when the bill is effective. **Note:** This column appears when you search for a fully-insured group bill.
End Date	Displays the date till when the bill is effective.

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
	Note: This column appears when you search for a fully-insured group bill.
Amount	Displays the bill amount.
	Note: This column appears when you search for a fully-insured group bill.
Status	Indicates the status of the bill. The valid values are: • Pending • Completed
	Note: This column appears when you search for a fully-insured group bill.
Billable Charge Information	Displays information about the bill.
	Note: It has a link. On clicking the link, the **Billable Charge ID** screen appears where you can view the details of the respective billable charge.
	Note: This column appears when you search for a fully-insured group bill.
Pricing Calculation ID	Displays the pricing calculation ID of the bill.
	Note: This column appears when you search for a fully-insured group bill.
Request Reference ID	Displays the request reference ID of the bill.
	Note: This column appears when you search for a fully-insured group bill.
Transaction ID	Displays the transaction ID of the bill.
	Note: This column appears when you search for a self-funded bill.
Transaction Date	Displays the date and time when the transaction was performed.
	Note: This column appears when you search for a self-funded bill.
Transaction Volume	Displays the transaction volume.
	Note: This column appears when you search for a self-funded bill.
Transaction Amount	Displays the transaction amount.
	Note: This column appears when you search for a self-funded bill.
Transaction Currency	Displays the currency in which the transaction was performed.
	Note: This column appears when you search for a self-funded bill.
Transaction Source	Displays the transaction source from where the transaction was received.
	Note: This column appears when you search for a self-funded bill.
Transaction Record Type	Displays the record type of the transaction.

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
	Note: This column appears when you search for a self-funded bill.
Initial Price Item	Indicates the price item to which the transaction is mapped.
	Note: This column appears when you search for a self-funded bill.
Rule	Indicates the rule which was met while determining initial price item for the transaction.
	Note: It has a link. On clicking the link, the **Rule** screen appears where you can view the details of the respective rule. This column appears when you search for a self-funded bill.
Transaction Information	Displays information about the transaction.
	Note: It has a link. On clicking the link, the **Transaction Details** screen appears where you can view the details of the respective transaction. This column appears when you search for a self-funded bill.

Related Topics

For more information on...	See...
How to search for a bill	Searching for a Bill on page 281
How to view the bill segments of a bill	Viewing the Bill Segments of a Bill on page 284

Bill Segments

The **Bill Segments** zone lists the bill segments of a bill. This zone contains the following columns:

Column Name	Column Description
Bill Segment ID	Displays the bill segment ID.
	Note: It has a link. On clicking the link, the **Bill Segment** screen appears with the details of the respective bill segment.
Status	Indicates the status of the bill segment.
Start Date	Displays the start date of the bill segment.
End Date	Displays the end date of the bill segment.
Amount	Displays the amount of the bill segment.
Currency	Indicates the currency in which the bill segment is generated.
Price Item	Indicates the price item for which the bill segment is generated.

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
Price Item Parameters	Displays a comma-separated list of parameter values. It indicates the price item parameters and their values which are used along with the price item for determining price item pricing.
	Note: This column appears when the **Multi Price Parameter** option type of the **Multi Parameter Based Pricing (C1_PPARM_FLG)** feature configuration is set to **True**.
Rate Schedule	Indicates the rate schedule which is used for defining the price item pricing.
Billable Charge ID	Indicates the billable charge for which the bill segment is generated.
	Note: It has a link. On clicking the link, the **Billable Charge** screen appears where you can view the details of the respective billable charge.
Account Information	Indicates the member account (in the invoicing group) for which the bill segment is generated.
Construct ID	Indicates the construct through which the bill segment is generated.

By default, the **Bill Segments** zone does not appear in the **Bill Drill Down Details** screen. It appears when you click the **Broadcast** (📶) icon corresponding to the bill in the **Search Billing Element** zone.

Note: Pagination is used to display limited number of records in this zone. You can use the navigation links, such as **Previous** and **Next** to navigate between pages.

You can filter the list using various search criteria (such as, **Bill Segment Start Date**, **Price Item**, and **Bill Segment End Date**) available in the **Filter** area. By default, the **Filter** area is hidden. You can view the **Filter** area by clicking the **Filters** (▽) icon in the upper right corner of this zone.

Related Topics

For more information on...	See...
How to search a bill	Searching for a Bill on page 281
How to view the bill segments of a bill	Viewing the Bill Segments of a Bill on page 284

SQ Details

The **SQ Details** zone lists the service quantity identifiers (SQIs) used in the bill segment calculation. It appears when the bill segment is generated for an SQI based billable charge. It contains the following columns:

Column Name	Column Description
SQI	Indicates the service quantity identifier.
Billed Service Quantity	Displays the service quantity used by the rates engine for calculation. This amount differs from the initial service quantity when the rate's SQ rules manipulate the service quantity.

Copyright © 2024, Oracle and/or its affiliates.

Bill Segment Calculation Lines

The **Bill Segment Calculation Lines** zone lists the calculation lines of the bill segment. It contains the following columns:

Column Name	Column Description
Header Sequence	Indicates the sequence in which the calculation line is generated.
Sequence	Ideally, the sequence would be 1 unless multiple versions of the rate were effective during the bill period. If there is more than one effective rate version, there will be a separate set of calculation lines for each version applied. The lines that apply to the first part of the bill period would have a Sequence of 1, the lines for the next part of the bill period would have a Sequence of 2 and so on.
Description on Bill	Displays the summarized information about the charge that will be printed on the bill.
Amount	Displays the charge amount.
Distribution Code	Displays the distribution code which indicates the GL account to which the charge must be posted.
Service Quantity	Indicates the service quantity which is used for calculation.

By default, the **Bill Segment Calculation Lines** zone does not appear in the **Bill Drill Down Details** screen. It appears when you click the **Broadcast** (ᯤ) icon corresponding to a bill segment in the **Bill Segments** zone.

Note: Pagination is used to display limited number of records in this zone. You can use the navigation links, such as **Previous** and **Next** to navigate between pages.

Timeline and Billable Charge Details

The **Timeline and Billable Charge Details** zone lists the timeline and billable charge details of a bill segment. This zone contains the following columns:

Column Name	Column Description
Start Date	Displays the date from when the timeline is effective for the bill segment.
End Date	Displays the date till when the timeline is effective for the bill segment.
Amount	Displays the billable charge amount calculated for the timeline.
Status	Indicates the status of the timeline.
Billable Charge Information	Displays the billable charge information of the bill segment. **Note:** It has a link. On clicking the link, the **Billable Charge ID** screen appears where you can view the details of the billable charge.
Pricing Calculation ID	Displays the pricing calculation ID of the bill segment.
Request Reference ID	Displays the repricing request reference ID of the bill segment.

By default, the **Timeline and Billable Charge Details** zone does not appear in the **Bill Drill Down Details** screen. It appears when you click the **Broadcast** (ᯤ) icon corresponding to the bill segment ID in the **Bill Segments** zone.

Copyright © 2024, Oracle and/or its affiliates.

Note: Pagination is used to display limited number of records in this zone. You can use the navigation links, such as **Previous** and **Next** to navigate between pages.

You can filter the list using various search criteria (such as, **Start Date**, **Billable Charge ID**, **End Date**, and **Status**) available in the **Filter** area. By default, the **Filter** area is hidden. You can view the **Filter** area by clicking the **Filters** (▽) icon in the upper right corner of this zone.

Related Topics

For more information on...	See...
How to view the timeline and billable charge details of a bill segment	Viewing the Timeline and Billable Charge Details of a Bill Segment on page 286

Audit Event Details

The **Audit Event Details** - zone lists the audit event. It contains the following columns:

Column Name	Column Description
Creation Date Time	Displays the date and time when the audit event is created.
Audit Event Information	Indicates the audit event type using which the audit event is created.
	Note: It has a link. On clicking the link, the **Audit Event** screen appears where you can view the details of the respective audit event.
Entity ID	Indicates the entity for which the audit event is created.
Entity	Indicates the entity business object for which the audit event is created.
Effective Date	Displays the effective date which is stamped corresponding to the audit event.
	Note: The effective date is used later to derive the timeline for premium calculation.
Status	Indicates the status of the audit event. The valid values are: • Canceled • Complete • Error • Pending
Audit Action	Indicates the action due to which the audit event is created for the entity. The valid values are: • Add Row • Change Row • Delete Row

By default, the **Audit Event Details** zone does not appear in the **Bill Drill Down Details** screen. It appears when you click the **Broadcast** (📶) icon corresponding to the timeline in the **Timeline and Billable Charge Details** zone.

Note: Pagination is used to display limited number of records in this zone. You can use the navigation links, such as **Previous** and **Next** to navigate between pages.

Copyright © 2024, Oracle and/or its affiliates.

You can filter the list using various search criteria (such as, **Created From**, **From Effective Date**, **Created To**, **To Effective Date**, and **Status**) available in the **Filter** area. By default, the **Filter** area is hidden. You can view the **Filter** area by clicking the **Filters** (▽) icon in the upper right corner of this zone.

Related Topics

For more information on...	See...
How to trace the source of a fully-insured group bill	Tracing the Source of a Fully-Insured Group Bill on page 284
How to view the audit event details of a bill	Viewing the Audit Event Details of a Timeline on page 286

Pricing Calculation Details

The **Pricing Calculation Details** - zone lists the calculation lines of the timeline and the billable charge. It contains the following columns:

Column Name	Column Description
Entity Information	Displays information about the entity.
Description on Bill	Displays the description of the bill.
Pricing Parameter	Indicates the price item parameters which are used along with the price item for determining the price item pricing.
Amount	Displays the pricing amount.
Total Amount	Displays the total amount of the entity.

Related Topics

For more information on...	See...
How to trace the source of a fully-insured group bill	Tracing the Source of a Fully-Insured Group Bill on page 284
How to view the pricing calculation details of a bill	Viewing the Pricing Calculation Details of a Timeline on page 287

Transaction Details

The **Transaction Details** zone lists the transactions whose charges are billed through the bill segment. It contains the following columns:

Column Name	Column Description
Transaction ID	Displays the transaction ID. **Note:** It has a link. On clicking the link, the **Transaction Details** screen appears where you can view the details of the respective transaction.
Transaction Date	Displays the date and time when the transaction was performed.
Transaction Volume	Displays the transaction volume.
Transaction Amount	Displays the transaction amount.

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
Transaction Currency	Displays the currency in which the transaction was performed.
Transaction Source	Displays the transaction source from where the transaction was received.
Transaction Record Type	Displays the record type of the transaction.
Initial Price Item	Indicates the price item to which the transaction is mapped.
Rule	Indicates the rule which was met while determining initial price item for the transaction.
	Note: It has a link. On clicking the link, the **Rule** screen appears where you can view the details of the respective rule.
Transaction Information	Displays information about the transaction.
	Note: It has a link. On clicking the link, the **Transaction Details** screen appears where you can view the details of the respective transaction.

Transaction Leg Calculation Lines

The **Transaction Leg Calculation Lines** zone lists the calculation lines of all legs of the transaction which are rated before billing. It contains the following columns:

Column Name	Column Description
Transaction Calculation ID	Displays the transaction calculation ID.
Sequence	Indicates the sequence in which the calculation line was created. It is the sequence of the rate component which is used in the rate schedule.
Price Component ID	Indicates the price component which is used during calculation.
Rate	Indicates the rate which is used for the calculation.
SQI	Indicates the service quantity identifier (SQI) which is used in the calculation.
Billable Service Quantity	Indicates the service quantities used by the rates engine for calculation.
Calculated Amount	Displays the calculated charge amount which is rounded off (i.e. amount in precision). For example, 100. This amount appears in the invoice currency.
Precise Calculated Amount	Displays the calculated charge amount which is not rounded off (i.e. amount with precision and scale). For example, 98.67. This amount appears in the invoice currency.
Exempt Amount	Displays the amount of the calculated charge that the person does not have to pay due to tax exemption.
Base Amount	Displays the total amount derived from the cross-referenced line(s) that the current line then used to calculate the charge amount.
Pricing Amount	Displays the calculated charge amount which is not rounded off (i.e. amount with precision and scale). For example, 98.67. This amount appears in the pricing currency.
Exchange Rate	Indicates the exchange rate which is used when the invoice currency is different from the pricing currency.

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
Print	Indicates whether information about the calculation line must be printed on the person's bill. The valid values are: • Y • N
Description on Bill	Displays the summarized information about the charge that will be printed on the bill.
Distribution Code	Displays the distribution code which indicates the GL account to which the charge must be posted.
Pricing Currency	Indicates the currency in which the price item pricing is defined.
Aggregation Parameter Group ID	Indicates the aggregation parameter group using which the transaction legs are aggregated during calculation.

Searching for a Bill

Procedure

To search for a bill:

1. Click the **Menu** link in the **Application** toolbar.
   A list appears.
2. From the **Main** menu, select **Billing Management** and then click **Bill Drill Down Details**.
   The **Bill Drill Down Details** screen appears.
3. Enter the search criteria in the **Search Billing Element** zone.

   Note: ORMB search engine supports wildcard search, where you can substitute the percentage (%) symbol as a stand in for any word or letter in a search criteria. You can use the '%' wildcard character in all input fields except the date and ID fields. The '%' wildcard character is suffixed automatically at the end of the partial search criteria. Therefore, you may or may not specify the wildcard character at the end of the partial search criteria. However, you have to prefix the wildcard character manually wherever required.
4. Click **Search**.
   A list of bills that meet the search criteria appears in the **Search Results** section.

Related Topics

For more information on...	See...
Bill Drill Down Details screen	Bill Drill Down Details on page 271
Search Billing Element zone	Search Billing Element on page 271

Tracing the Source of a Bill (Non-TFM)

Procedure

To trace the source of a non-tfm bill:

1. Select the **Bill Details** option from the **Search By** field in the **Bill Drill Down Details** screen.
2. Search for the non-tfm bill in the **Bill Drill Down Details** screen.

Copyright © 2024, Oracle and/or its affiliates.

3. In the **Search Results** section, click the **Broadcast** (ᯤ) icon corresponding to the non-tfm bill whose source you want to trace.
   The **Bill Segments** zone appears.
4. View the details of the bill segments in the **Bill Segments** zone.
5. In the **Bill Segments** zone, click the **Broadcast** (ᯤ) icon corresponding to the bill segment whose details you want to view.
   The **SQ Details**, **Bill Segment Calculation Lines**, and **Timeline and Billable Charge Details** zones appear.
6. View the SQ details of the bill segment in the **SQ Details** zone.
7. View the calculation lines of the bill segment in the **Bill Segment Calculation Lines** zone.
8. View the timeline and billable charge details of the bill segment in the **Timeline and Billable Charge Details** zone.
9. If required, you can view the pricing and audit event details of a timeline, by clicking the **Broadcast** (ᯤ) icon corresponding to the timeline in the **Timeline and Billable Charge Details** zone.
   The **Pricing Calculation Details** and **Audit Event Details** zone appears.
10. If required, view the pricing calculation details of a timeline in the **Pricing Calculation Details** zone.
11. If required, view the audit event details of a timeline in the **Audit Event Details** zone.

Related Topics

For more information on...	**See...**
Bill Drill Down Details screen	Bill Drill Down Details on page 271
Bill Segments zone	Bill Segments on page 275
SQ Details zone	SQ Details on page 276
Bill Segment Calculation Lines zone	Bill Segment Calculation Lines on page 277
Timeline and Billable Charge Details zone	Timeline and Billable Charge Details on page 277
Pricing Calculation Details zone	Pricing Calculation Details on page 279
Audit Event Details zone	Audit Event Details on page 278

Tracing the Source of a Bill Generated Through TFM

Procedure

To trace the source of a bill generated through TFM:

1. Select the **Bill Details** option from the **Search By** field in the **Bill Drill Down Details** screen.
2. Search for the bill (generated through the TFM process) in the **Bill Drill Down Details** screen.
3. In the **Search Results** section, click the **Broadcast** (ᯤ) icon corresponding to the bill whose source you want to trace.
   The **Bill Segments** zone appears.
4. View the details of the bill segments in the **Bill Segments** zone.
5. In the **Bill Segments** zone, click the **Broadcast** (ᯤ) icon corresponding to the bill segment whose details you want to view.
   The **SQ Details**, **Bill Segment Calculation Lines**, and **Transaction Details** zones appear.
6. View the SQ details of the bill segment in the **SQ Details** zone.
7. View the calculation lines of the bill segment in the **Bill Segment Calculation Lines** zone.
8. View the transaction details of the bill segment in the **Transaction Details** zone.
9. If required, you can view the calculation lines of a transaction leg, by clicking the **Broadcast** (ᯤ) icon corresponding to the transaction in the **Transaction Details** zone.

Copyright © 2024, Oracle and/or its affiliates.

The **Transaction Leg Calculation Lines** zone appears.

10. If required, view the calculation lines of a transaction leg in the **Transaction Leg Calculation Lines** zone.

Related Topics

For more information on...	See...
Bill Drill Down Details screen	Bill Drill Down Details on page 271
Bill Segments zone	Bill Segments on page 275
SQ Details zone	SQ Details on page 276
Bill Segment Calculation Lines zone	Bill Segment Calculation Lines on page 277
Transaction Details zone	Transaction Details on page 279
Transaction Leg Calculation Lines zone	Transaction Leg Calculation Lines on page 280

Tracing the Source of a Self-Funded Bill

Procedure

To trace the source of a self-funded bill:

1. Select the **Bill Details** option from the **Search By** field in the **Bill Drill Down Details** screen.
2. Search for a self-funded bill in the **Bill Drill Down Details** screen.
3. In the **Search Results** section, click the **Broadcast** (ᯤ) icon corresponding to the bill whose source you want to view.
   The **Bill Segments** zone appears.
4. View the details of the bill segments in the **Bill Segments** zone.
5. In the **Bill Segments** zone, click the **Broadcast** (ᯤ) icon corresponding to the bill segment whose details you want to view.
   The **SQ Details**, **Bill Segment Calculation Lines**, and **Transaction Details** zones appear.
6. View the SQ details of the bill segment in the **SQ Details** zone.
7. View the calculation lines of the bill segment in the **Bill Segment Calculation Lines** zone.
8. View the transaction details of the bill segment in the **Transaction Details** zone.
9. If required, you can view the transaction leg calculation lines of a transaction, by clicking the **Broadcast** (ᯤ) icon corresponding to the transaction in the **Transaction Details** zone.
   The **Transaction Leg Calculation Lines** zone appears.
10. View the transaction leg calculation lines of a transaction in the **Transaction Leg Calculation Lines** zone.

Related Topics

For more information on...	See...
Bill Drill Down Details screen	Bill Drill Down Details on page 271
Bill Segments zone	Bill Segments on page 275
SQ Details zone	SQ Details on page 276
Bill Segment Calculation Lines zone	Bill Segment Calculation Lines on page 277
Transaction Details zone	Transaction Details on page 279
Transaction Leg Calculation Lines zone	Transaction Leg Calculation Lines on page 280

Copyright © 2024, Oracle and/or its affiliates.

Tracing the Source of a Fully-Insured Group Bill

Procedure

To trace the source of a fully-insured group bill:

1. Select the **Bill Details** option from the **Search By** field in the **Bill Drill Down Details** screen.
2. Search for a fully-insured group bill in the **Bill Drill Down Details** screen.
3. In the **Search Results** section, click the **Broadcast** (📶) icon corresponding to the bill whose source you want to trace.
   The **Bill Segments** zone appears.
4. View the details of the bill segments in the **Bill Segments** zone.
5. In the **Bill Segments** zone, click the **Broadcast** (📶) icon corresponding to the bill segment whose details you want to view.
   The **SQ Details**, **Bill Segment Calculation Lines**, and **Timeline and Billable Charge Details** zones appear.
6. View the SQ details of the bill segment in the **SQ Details** zone.
7. View the calculation lines of the bill segment in the **Bill Segment Calculation Lines** zone.
8. View the timeline and billable charge details of the bill segment in the **Timeline and Billable Charge Details** zone.
9. If required, you can view the pricing and audit event details of a timeline, by clicking the **Broadcast** (📶) icon corresponding to the timeline in the **Timeline and Billable Charge Details** zone.
   The **Pricing Calculation Details** and **Audit Event Details** zone appears.
10. If required, view the pricing calculation details of a timeline in the **Pricing Calculation Details** zone.
11. If required, view the audit event details of a timeline in the **Audit Event Details** zone.

Related Topics

For more information on...	**See...**
Bill Drill Down Details screen	Bill Drill Down Details on page 271
Bill Segments zone	Bill Segments on page 275
SQ Details zone	SQ Details on page 276
Bill Segment Calculation Lines zone	Bill Segment Calculation Lines on page 277
Timeline and Billable Charge Details zone	Timeline and Billable Charge Details on page 277
Pricing Calculation Details zone	Pricing Calculation Details on page 279
Audit Event Details zone	Audit Event Details on page 278

Viewing the Bill Segments of a Bill

Procedure

To view the bill segments of a bill:

1. Search for the bill in the **Bill Drill Down Details** screen.
2. In the **Search Results** section, click the **Broadcast** (📶) icon corresponding to the bill whose bill segments you *w*ant to view.
   The **Bill Segments** zone appears.
3. View all bill segments of the bill in the **Bill Segments** zone.

Related Topics

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
How to search for a bill	Searching for a Bill on page 281
Bill Segments zone	Bill Segments

Viewing the SQ Details of a Bill Segment

Procedure

To view the SQ details of a bill segment:

1. Search for the bill, whose details you want to view, in the **Bill Drill Down Details** screen.
2. In the **Search Results** section, click the **Broadcast** (ᯤ) icon corresponding to the bill whose details you want *t*o view.
   The **Bill Segments** zone appears.
3. In the **Bill Segments** zone, click the **Broadcast** (ᯤ) icon corresponding to the bill segment whose details you want to view.
   The **SQ Details**, **Bill Segment Calculation Lines**, and **Transaction Details** zones appear.

   Note:

   The **SQ Details** zone appears when the bill segment is generated for an SQI based billable charge.

   The **Transaction Details** zone appears when the bill segment is generated for a billable charge which is created through the transaction aggregation process.
4. View SQ details of the bill segment in the **SQ Details** zone.

Related Topics

For more information on...	See...
How to search for a bill	Searching for a Bill on page 281
Bill Drill Down Details screen	Bill Drill Down Details on page 271
Bill Segment Details zone	Bill Segments on page 275
SQ Details zone	SQ Details on page 276

Viewing the Calculation Lines of a Bill Segment

Procedure

To view the calculation lines of a bill segment:

1. Search for the bill, whose details you want to view, in the **Bill Drill Down Details** screen.
2. In the **Search Results** section, click the **Broadcast** (ᯤ) icon corresponding to the bill whose details you want *t*o view.
   The **Bill Segments** zone appears.
3. In the **Bill Segments** zone, click the **Broadcast** (ᯤ) icon corresponding to the bill segment whose details you want to view.
   The **Bill Segment Calculation Lines** screen appears.

Copyright © 2024, Oracle and/or its affiliates.

> **Note:**
>
> The **SQ Details** zone appears when the bill segment is generated for an SQI based billable charge.
>
> The **Transaction Details** zone appears when the bill segment is generated for a billable charge which is created through the transaction aggregation process.

4. View the calculation lines of the bill segment in the **Bill Segment Calculation Lines** zone.

Related Topics

For more information on...	See...
How to search for a bill	Searching for a Bill on page 281
Bill Drill Down Details screen	Bill Drill Down Details on page 271
Bill Segments zone	Bill Segments on page 275
SQ Details zone	SQ Details on page 276
Bill Segment Calculation Lines zone	Bill Segment Calculation Lines on page 277

Viewing the Timeline and Billable Charge Details of a Bill Segment

Procedure

To view the timeline and billable charge details of a bill segment:

1. Search for the bill in the **Bill Drill Down Details** screen.
2. In the **Search Results** section, click the **Broadcast** (📶) icon corresponding to the *b*ill ID whose timeline and billable charge details you want to view.
   The **Bill Segments** zone appears.
3. In the **Bill Segments** zone, click the **Broadcast** (📶) icon corresponding to the *b*ill segment whose timeline and billable charge details you want to view.
   The **Timeline and Billable Charge Details** zone appears.
4. View the timeline and billable charge details of a bill segment in the **Timeline and Billable Charge Details** zone.

Related Topics

For more information on...	See...
How to trace the source of a fully-insured group bill	Tracing the Source of a Fully-Insured Group Bill on page 284
Bill Drill Down Details screen	Bill Drill Down Details on page 271
Bill Segment zone	Bill Segments on page 275
SQ Details zone	SQ Details on page 276
Bill Segment Calculation Lines zone	Bill Segment Calculation Lines on page 277
Timeline and Billable Charge Details zone	Timeline and Billable Charge Details on page 277

Viewing the Audit Event Details of a Timeline

Procedure

Copyright © 2024, Oracle and/or its affiliates.

To view the audit event details of a timeline:

1. Search for the fully-insured group bill in the **Bill Drill Down Details** screen.
2. In the **Search Results** section, click the **Broadcast** (📶) icon corresponding to the fully-insured group *b*ill ID whose bill segment details you want to view.
   The **Bill Segments** zone appears.
3. In the **Bill Segments** zone, click the **Broadcast** (📶) icon corresponding to the *b*ill segment whose timeline details you want to view.
   The **SQ Details**, **Bill Segment Calculation Lines**, and **Timeline and Billable Charge Details** zones appears.
4. In the **Timeline and Billable Charge Details** zone, click the **Broadcast** (📶) icon corresponding to the *t*imeline whose event details you want to view.
   The **Audit Event Details** and **Pricing Calculation Details** zone appears.
5. View the audit event details of a timeline in the **Audit Event Details** zone.

Related Topics

For more information on...	**See...**
How to trace the source of a fully-insured group bill	Tracing the Source of a Fully-Insured Group Bill on page 284
Bill Drill Down Details screen	Bill Drill Down Details on page 271
Bill Segment zone	Bill Segments on page 275
SQ Details zone	SQ Details on page 276
Bill Segment Calculation Lines zone	Bill Segment Calculation Lines on page 277
Timeline and Billable Charge Details zone	Timeline and Billable Charge Details on page 277
Audit Event Details zone	Audit Event Details on page 278

Viewing the Pricing Calculation Details of a Timeline

Procedure

To view the pricing calculation details of a timeline:

1. Search for the fully-insured group bill in the **Bill Drill Down Details** screen.
2. In the **Search Results** section, click the **Broadcast** (📶) icon corresponding to the fully-insured group *b*ill ID whose bill segment details you want to view.
   The **Bill Segments** zone appears.
3. In the **Bill Segments** zone, click the **Broadcast** (📶) icon corresponding to the *b*ill segment whose timeline details you want to view.
   The **SQ Details**, **Bill Segment Calculation Lines**, and **Timeline and Billable Charge Details** zones appears.
4. In the **Timeline and Billable Charge Details** zone, click the **Broadcast** (📶) icon corresponding to the *t*imeline whose event details you want to view.
   The **Event Details** and **Pricing Calculation Details** zone appears.
5. View the pricing calculation details of a timeline in the **Pricing Calculation Details** zone.

Related Topics

For more information on...	**See...**
How to trace the source of a fully-insured group bill	Tracing the Source of a Fully-Insured Group Bill on page 284
Bill Drill Down Details screen	Bill Drill Down Details on page 271

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
Bill Segment zone	Bill Segments on page 275
SQ Details zone	SQ Details on page 276
Bill Segment Calculation Lines zone	Bill Segment Calculation Lines on page 277
Timeline and Billable Charge Details zone	Timeline and Billable Charge Details on page 277
Audit Event Details zone	Audit Event Details on page 278
Pricing Calculation Details zone	Pricing Calculation Details on page 279

Viewing the Transactions for Which the Bill Segment is Created

Procedure

To view the transactions for which the bill segment is created:

1. Search for the self-funded bill in the **Bill Drill Down Details** screen.
2. In the **Search Results** section, click the **Broadcast** (ᯤ) icon corresponding to the self-funded bill whose details you want *t*o view.
   The **Bill Segments** zone appears.
3. In the **Bill Segments** zone, click the**Broadcast** (ᯤ) icon corresponding to the bill segment whose details you want to view.
   The **SQ Details**, **Bill Segment Calculation Lines**, and **Transaction Details** zones appear.

   Note:

   The **SQ Details** zone appears when the bill segment is generated for an SQI based billable charge.

   The **Transaction Details** zone appears when the bill segment is generated for a billable charge which is created through the transaction aggregation process.
4. View the transactions details in the **Transaction Details** zone.

Related Topics

For more information on...	See...
How to search for the source of a fully-insured group bill	Tracing the Source of a Fully-Insured Group Bill on page 284
Bill Drill Down Details screen	Bill Drill Down Details on page 271
Bill Segments zone	Bill Segments on page 275
SQ Details zone	SQ Details on page 276
Bill Segment Calculation Lines zone	Bill Segment Calculation Lines on page 277
Transaction Details zone	Transaction Details on page 279

Viewing the Transaction Leg Calculation Lines of the Transaction

Procedure

To view the transaction leg calculation lines of the transaction:

1. Search for the self-funded bill in the **Bill Drill Down Details** screen.

Copyright © 2024, Oracle and/or its affiliates.

2. In the **Search Results** section, click the **Broadcast** (ᯤ) icon corresponding to the self-funded bill whose details you want *t*o view.
   The **Bill Segments** zone appears.
3. In the **Bill Segments** zone, click the **Broadcast** (ᯤ) icon corresponding to the bill segment whose details you want to view.
   The **SQ Details**, **Bill Segment Calculation Lines**, and **Transaction Details** zones appear.

   Note:

   The **SQ Details** zone appears when the bill segment is generated for an SQI based billable charge.

   The **Transaction Details** zone appears when the bill segment is generated for a billable charge which is created through the transaction aggregation process.

4. In the **Transaction Details** zone, click the **Broadcast** (ᯤ) icon corresponding to the transaction whose leg calculation lines you want to view.
   The **Transaction Leg Calculation Lines** zone appears.
5. View the calculation lines of all the legs which are rated before billing in the **Transaction Leg Calculation Lines** zone.

Related Topics

For more information on...	**See...**
How to trace for the source of a self-funded bill	Tracing the Source of a Self-Funded Bill on page 283
Bill Drill Down Details screen	Bill Drill Down Details on page 271
Bill Segments zone	Bill Segments on page 275
SQ Details zone	SQ Details on page 276
Bill Segment Calculation Lines zone	Bill Segment Calculation Lines on page 277
Transaction Details zone	Transaction Details on page 279
Transaction Legs Calculation Lines zone	Transaction Leg Calculation Lines on page 280

Tracing the Source of a Billable Charge Generated Through TFM

Procedure

To trace the source of a billable charge generated through TFM:

1. Select the **Billable Charge Details** option from the **Search By** field in the **Bill Drill Down Details** screen.
2. Search for a billable charge generated through TFM in the **Bill Drill Down Details** screen.
3. In the **Search Results** section, click the **Broadcast** (ᯤ) icon corresponding to the transaction whose calculation lines you want to view.
   The **Transaction Leg Calculation Lines** zone appears.
4. View the leg calculation lines of a transaction in the **Transaction Leg Calculation Lines** zone.

Related Topics

For more information on...	**See...**
Bill Drill Down Details screen	Bill Drill Down Details on page 271
Transaction Leg Calculation Lines zone	Transaction Leg Calculation Lines on page 280

Copyright © 2024, Oracle and/or its affiliates.

Tracing the Source of a Self-Funded Billable Charge

Procedure

To trace the source of a self-funded billable charge:

1. Select the **Billable Charge Details** option from the **Search By** field in the **Bill Drill Down Details** screen.
2. Search for a self-funded billable charge in the **Bill Drill Down Details** screen.
3. In the **Search Results** section, click the **Broadcast** (📶) icon corresponding to the transaction whose leg calculation lines you want to view.
   The **Transaction Leg Calculation Lines** zone appears.
4. View the leg calculation lines of a transaction in the **Transaction Leg Calculation Lines** zone.

Related Topics

For more information on...	See...
Bill Drill Down Details screen	Bill Drill Down Details on page 271
Transaction Leg Calculation Lines zone	Transaction Leg Calculation Lines on page 280

Tracing the Source of a Fully-Insured Group Billable Charge

Procedure

To trace the source of a fully-insured group billable charge:

1. Select the **Billable Charge Details** option from the **Search By** field in the **Bill Drill Down Details** screen.
2. Search for a fully-insured group billable charge in the **Bill Drill Down Details** screen.
3. In the **Search Results** section, click the **Broadcast** (📶) icon corresponding to the billable charge whose details you want to view.
   The **Pricing Calculation Details** and **Audit Event Details** zone appears.
4. View the pricing calculation details of a billable charge in the **Pricing Calculation Details** zone.
5. View the audit event details of a billable charge in the **Audit Event Details** zone.

Related Topics

For more information on...	See...
Bill Drill Down Details screen	Bill Drill Down Details on page 271
Pricing Calculation Details zone	Pricing Calculation Details on page 279
Audit Event Details zone	Audit Event Details on page 278

Viewing the Bill Segments of a Billable Charge

Procedure

To view the bill segments of a billable charge:

1. Ensure that you select the **Billable Charge Details** option from the **Search By** field in the **Bill Drill Down Details** screen.

Copyright © 2024, Oracle and/or its affiliates.

2. Enter the billable charge ID as the search criteria in the **Billable Charge ID** field.
   The output listed in the **Search Results** section vary depending on the following scenarios:

If...	Then...
The billable charge ID belongs to a fully-insured group bill	**a.** Search for the bill in the **Bill Drill Down Details** screen. **b.** In the **Search Results** section, click the **Broadcast** (ᯤ) icon corresponding to the billable charge whose bill segments you want to view. The **Pricing Calculation Details** and **Audit Event Details** zone appear. **c.** View the details of the bill segment in the **Pricing Calculation Details** and **Audit Event Details** zones.
The billable charge ID belongs to a self-funded bill	**a.** Search for the bill in the **Bill Drill Down Details** screen. **b.** In the **Search Results** section, click the **Broadcast** (ᯤ) icon corresponding to the transaction whose transaction leg calculation lines you want to view. The **Transaction Leg Calculation Lines** zone appear. **c.** View the leg calculation lines of the transaction in the **Transaction Leg Calculation Lines** zone.

Related Topics

For more information on...	See...
Pricing Calculation Details zone	Pricing Calculation Details on page 279
Audit Event Details zone	Audit Event Details on page 278
Transaction Leg Calculation Lines zone	Transaction Leg Calculation Lines on page 280

Alert Notification for Bill and Bill Segment Errors

Oracle Revenue Management and Billing enables you to generate alert notifications for bills and bill segments which are in the **Error** status. At present, this feature considers those bills and bill segments which are generated through a batch process and not those which are generated manually from the user interface. The system supports the following two alert notifications:

- To Do Creation
- Email Notification

The objective of these notifications is to create an alert for the bills and bill segments which are in the **Error** status. Once a To Do is created, it indicates the number of bills and bill segments which are in the **Error** status on the batch business date. However, the email notification provides the following information:

- **Bill Error Summary Statistics** - Indicates the number of bills where a particular error has occurred in each bill cycle.
- **Bill Segment Error Summary Statistics** - Indicates the number of bill segments where a particular error has occurred in each bill cycle.

The **C1-ERRAL** batch is enhanced to generate alert notifications for bills and bill segments which are in the **Error** status. While executing the **C1-ERRAL** batch, you need to set the **To Do Created For** parameter to **BILL**. The system then checks whether there are any bills or bill segments which are currently in the **Error** status. If there are any bills or bill segments in the **Error** status, the system creates a To Do using the **C1-BLLER** To Do type (which is shipped with the product). Note that the system will not create a new To Do when the To Do for the same drill key combination already exists in the **Open** status.

Copyright © 2024, Oracle and/or its affiliates.

To create an email notification, you need to attach an algorithm which is created using the **C1-BLLEREML** algorithm type to the **To Do Post-Processing** system event of the **C1-BLLER** To Do type. On creating a To Do, the batch then creates an email outbound message which contains the error summary statistics for bills and bill segments.

Chapter 3

Payments

Topics:

- The Big Picture of Payments
- Maintaining Payment Events
- Payment Event QuickAdd
- Payment Quick Add
- Maintaining Payments
- Payment Portal
- How To
- Financial Transactions On A Payment
- Payment History
- Remittance Summary
- Account Payment History
- Payment Tender History
- Payment / Tender Search
- Payment Event Exception
- Payment Exception
- Maintaining Deposit Controls
- Maintaining Tender Controls
- Interfacing Payments From External Sources
- Payment Creation, Cancellation, Transfer, and View Through Inbound Web Service

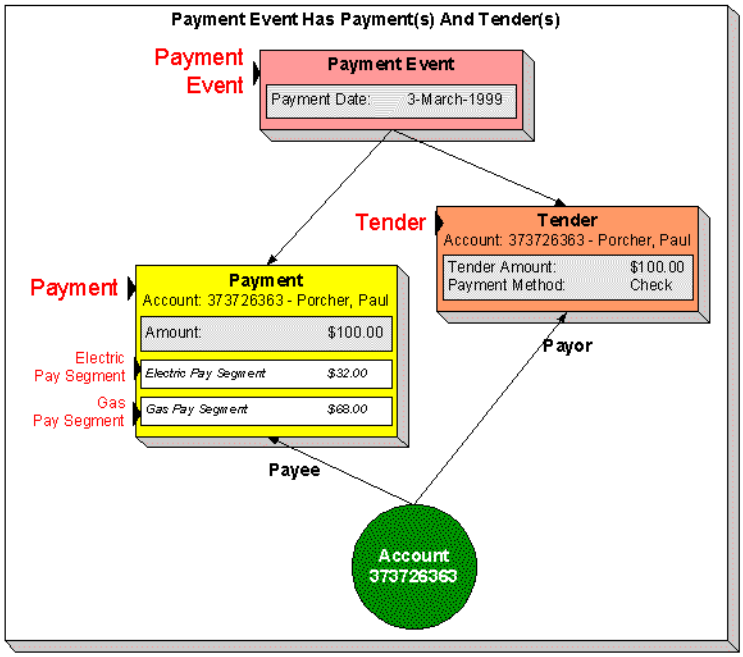
In this section, we describe how to manage your customer's payments.

The Big Picture of Payments

A payment reduces how much an account owes. The topics in this section provide background information about a variety of payment topics.

A Payment Event Has Payments And Tenders

The explanation in The Financial Big Picture provides an accurate, but incomplete view of payments. The missing pieces concern *payment events* and *tenders*. The following diagram illustrates the difference between a payment event, its payment(s) and its tender(s).



The following concepts are illustrated above:

A payment event defines the event A payment event is required whenever any form of payment is received. The payment event defines the payment date (and that's all).

A payment event has tender(s) A tender exists for every form of tender remitted as part of the payment event. A payment event must have at least one tender otherwise nothing was remitted. A payment event may have many tenders when multiple payment methods are associated with an event (e.g., paying with cash, a check, and a credit card).

A payment is allocated to account(s) The total amount of tenders under a payment event is distributed to one or more accounts.

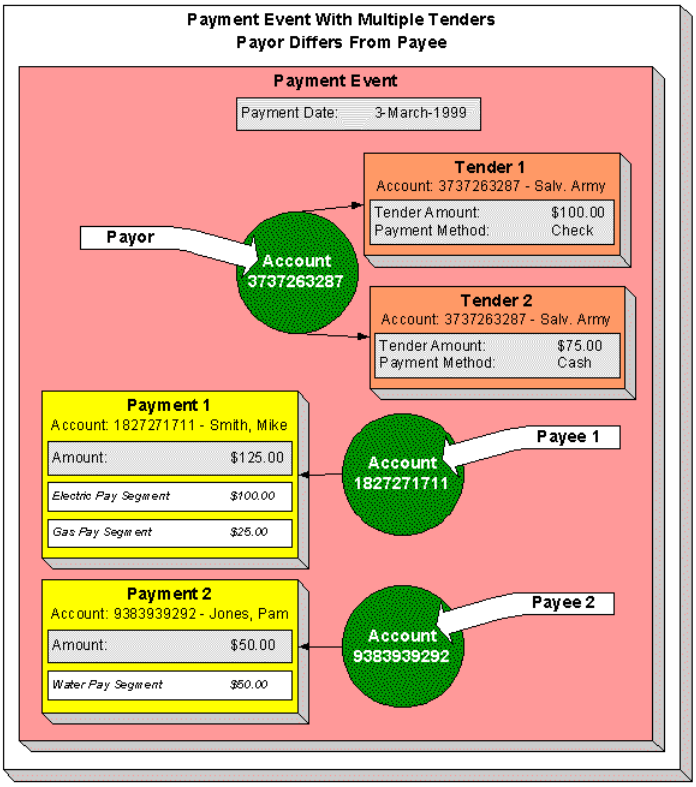
A payment is distributed to contract's The system allocates an account's payment amount amongst its contracts. The system creates a payment segment for each contract that receives a portion of the payment.

Payor and payee are frequently the same The account remitting the tender (the payor) is frequently the same as the account to which the funds are allocated (the payee). The next illustration provides an example when this is not the case.

Multiple Tenders Used To Pay For Multiple Accounts

The following diagram illustrates a payment event with multiple tenders where the payor of the tender is not the same as the account(s) receiving the payment.

Copyright © 2024, Oracle and/or its affiliates.



A payment event may have many tenders A single payment event may have many tenders. While the above example shows both tenders being paid by the same account, each tender may reference a different account.

Many accounts may be paid under 1 event The total amount of tenders under a payment event are distributed to one or more accounts.

Payor may differ from payee The account(s) remitting the tender may differ from the account(s) whose debt is relieved.

An Overview Of The Payment Event Creation & Allocation Process

When a payment event occurs, the system stores a tender for each form of remittance (e.g., cash, check, charge). It then allocates the sum of the tenders to one or more accounts.

By default, the system allocates the sum of the tenders to the account that remits the tenders. You may override this default and specify any number of accounts and their respective payment allocation amount. This is useful, for example, when a social service agency pays for many accounts. If applicable, you may also configure the system to use your own payment event distribution rule(s).

Fastpath: Refer to Distributing A Payment Event for more information.

The system distributes a payment amongst an account's contracts based on the age of each contract's debt AND distribution priority. The system creates a payment segment for each contract that receives part of the payment.

Fastpath: Refer to Distributing A Payment Amongst An Account's Contracts for more information.

You may manually redistribute the payment amount amongst the account's contracts before you commit the distribution. When the distribution is acceptable, you freeze the payment. Freezing a payment causes the system to create a financial transaction for each related payment segment. It is the financial transaction(s) that causes the contracts' payoff and current balances to be reduced. The financial transaction also contains the journal details that debit "cash" and credit some other GL account.

Copyright © 2024, Oracle and/or its affiliates.

And that's it. The remaining topics in this section provide more information about the creation and allocation of payment events.

Note: Batch and real-time payment event creation / allocation. There is only one payment event creation / allocation routine and therefore anything the batch payment process does for whole batches of payments, you can do to a payment on-line.

Distributing A Payment Event

CAUTION: This section deals with the concept distributing a payment amount into payment(s). It does not discuss the distribution of a single payment into segments. **For more information** about payment distribution, refer to Distributing A Payment Amongst An Account's Contracts.

The base-package, by default, creates a single payment for a payment event. Some business practices require potentially many payments to be created when payment events are added.

A few examples of when multiple payments may be necessary are:

- A payment amount needs to be distributed towards different distribution types:
  - $50 in interest
  - $60 in collection charges
  - $70 in taxes
- Each of the above distributions is realized as a separate payment identified by its own match type (i.e., there'll be one match type called Interest, another called Collection Charges, etc.).

In this example, the debt of a single contract may be relieved by each of these payments.

- In a similar way, you may want to create a separate payment for an overpayment to differentiate it from regular payments (using yet another match type).
- In the case of a social service agency that pays for many accounts, a single payment event may be distributed amongst multiple accounts.

The method by which a payment amount is distributed to create payment(s) is contained in `Create Payment` algorithms plugged in on a distribution rule.

There is yet another aspect to having control over how payment events are created. The default method of creating payment events assumes knowledge of account IDs (of the payor and the payee) when making a payment. In cases where payments are made by and towards business entities other than accounts, knowledge of their corresponding account IDs may not be available at payment time. Consider the following examples:

- A payment is made to pay a person's bill, providing the person's identification at payment time.
- A payment is made towards a specific contract using a "contract number" defined as contract characteristic.

The method by which the tender account is determined by means of an alternate identifier is contained in `Determine Tender Account` algorithm plugged in on a distribution rule.

Fastpath: Refer to Making Payments Using Distribution Rules for information on how to configure your system to use this distribution method.

Distributing A Payment Amongst An Account's Contracts

CAUTION: This section deals with the concept of distributing a payment amongst an account's contracts. It does not discuss how the sum of a payment event's tenders is balanced out by payment allocations. **For more information** about payment event balance, refer to Unbalanced Payment Events.

Copyright © 2024, Oracle and/or its affiliates.

A payment must be distributed to one or more contracts for its financial impact to be realized. When a payment satisfies an account's entire debt, you don't have to worry about how the system distributes the payment. The concept of payment distribution is only relevant when a partial or excess payment is distributed.

The first important point to understand is that the method of distributing a payment amongst an account's contracts is contained in an algorithm that's plugged in on to Customer Class. This means that you can have different distribution algorithms for different customer classes.

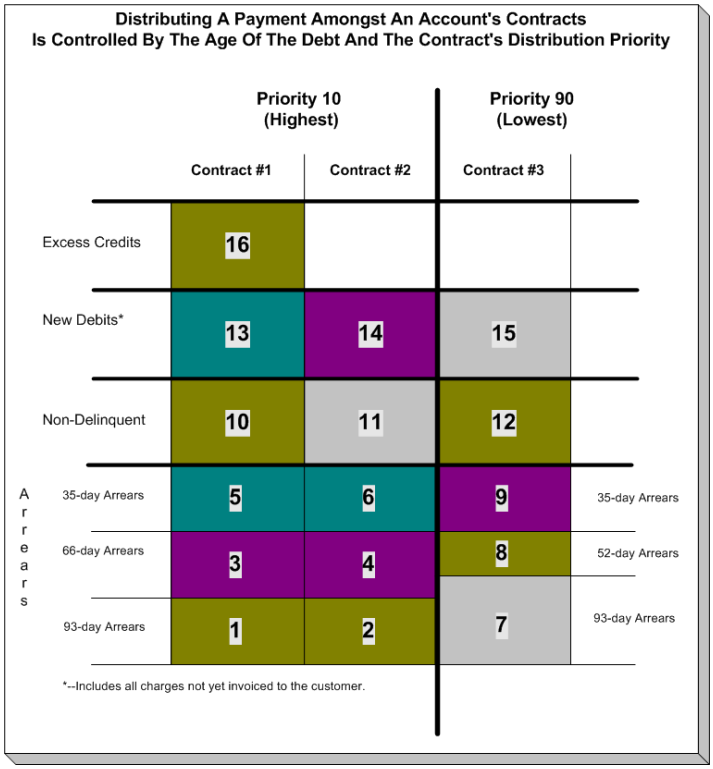
Note: Manual overrides. Most of the time, you'll let the payment distribution algorithm distribute the payment amongst an account's contracts. However, you may manually distribute a payment when a customer directs a payment to specific contract(s).

The following explanation describes one of the base package payment distribution algorithms (refer to Payment Distribution - Pay Priority and Debt Age for information about this algorithm). This algorithm distributes a payment based on:

- The age of each contract's debt.
- The payment distribution priority of each contract's contract type.

The following diagram helps illustrate how the distribution algorithm works.

Note: Important! There are other payment distribution algorithms in the base package. Click here to see the available algorithm types.



The above example shows three columns, one for each contract linked to a hypothetical account. Notice that two of the contracts have the same distribution priority, the third has a lower priority. The numbers in the cells indicate the order in which the system distributes a partial payment.

Note: Debt terminology. Before we can discuss the distribution algorithm, you must understand the terminology we use to categorize debt. ***Delinquent*** debt is associated with financial transactions that appear on overdue bills. ***Non-delinquent*** debt is associated with financial transactions that appear on current bills. ***New debits*** debt is associated with financial transactions that do not yet appear on a completed bill (e.g., a late payment charge that hasn't been billed yet).

The following points describe the algorithm used to distribute the partial payment:

Copyright © 2024, Oracle and/or its affiliates.

- The system pays off delinquent debt of the highest priority contracts first. In the above example, where multiple contracts have the same distribution priority, the system does NOT payoff one contract before it starts on the next (which one would it pick?). Rather, it distributes the payment amongst the contracts based on the age of the respective debt on each contract. In the above example, this is represented by steps 1 through 6 (notice how the distribution jumps between Contract 1 and Contract 2).
- After all delinquent debt has been relieved from the highest priority contract(s), the system pays off the next priority until all delinquent debt is relieved. In the above example, this is represented by steps 7 through 9.
- The system next pays off non-delinquent debt using each contract's respective distribution priority. Note well, the payment distribution algorithm doesn't associate an age with non-delinquent debt and therefore the distribution is based purely on the contracts' respective distribution priority. In the above example, this is represented by steps 10 through 12.
- After all non-delinquent debt is relieved, the system next pays off "new debit" debt based on the contracts' respective distribution priority. In the above example, this is represented by steps 13 through 15.
- Refer to Overpayment for a description of what happens if money still exists after the above distribution is complete.

Note: Payment segments and financial transactions. A payment segment exists for each contract that receives a portion of a payment. Linked to each payment segment is a financial transaction. It is the financial transaction that causes the contract's debt to be relieved and the general ledger to be impacted.

Fastpath: Refer to Payment Exception for more information about how the system handles errors detected during the payment distribution process.

Note: Overriding the distribution algorithm for uploaded payments. The standard distribution algorithm is used for payments that are Interfaced From An External System unless you specify a MATCH_VALUE and MATCH_FLG on the Payment Staging row associated with the uploaded payment. These fields are used in conjunction to indicate that the distribution of the payment should be restricted in some way (i.e., the standard payment distribution algorithm should not be used). MATCH_FLG indicates how the payment should be distributed (e.g., only distribute to a specific contract), MATCH_VALUE contains the ID of the restriction (e.g., the Contract ID).

Note: Open item customers. For an open-item customer, you MUST override the standard distribution algorithm because the payment is distributed as per the open items that it is relieving. Refer to Payments And Match Events for more information.

Overpayment

Overpayment refers to the situation where money is left over after a payment has been distributed to all eligible contracts, and all debt is relieved. Refer to Overpayment Segmentation for a description on how to configure the system to handle your overpayment requirements.

Canceling A Tender Versus Canceling A Payment

A payment event has tender(s) and payment(s). You can cancel a tender when it's not valid, e.g., when a check bounces. You can cancel a payment when the account should not have received the payment (e.g., a misdistribution or a canceled tender).

When you cancel a tender, the system automatically cancels ALL `frozen` payments. We do this because if the tender is canceled, there are no funds to distribute to accounts (unless there are other non-canceled tenders under the event). However, when you cancel a payment, the system does NOT cancel the tender(s) because we assume that, if the tenders were incorrect, you would have canceled them rather than the payment.

Copyright © 2024, Oracle and/or its affiliates.

NSF Cancellations

When a tender is canceled, a cancellation reason must be supplied. If the cancellation reason indicates a NSF (non sufficient funds) charge should be levied, the system invokes the NSF charge algorithm specified on the tender's account's customer class. Algorithms of this type will typically create an adjustment or billable charge to levy the NSF charge. Refer to NSFC-DFLT for an example of such an algorithm.

> **Note: The payor gets the NSF adjustments.** It's possible for the payor of a tender to differ from the payee (e.g., when a social service agency pays for other customers' debt). When you cancel a tender you must specify a cancellation reason. If the cancellation reason indicates an NSF charge should be levied, the sample algorithm provided with the system levies the NSF charge on the PAYOR's account. Because adjustments must be linked to a contract, the algorithm picks the payor's contract with the highest payment distribution priority and levies the adjustment specified in the contract's contract type's NSF adjustment type.

Besides calling this algorithm, a NSF cancellation may affect the tendering account's credit rating and cash-only score. The cancellation reason indicates the extent to which the account's ratings are affected.

And finally, whenever a tender is canceled due to non-sufficient funds, the account is scheduled for review by the Account Debt Monitor.

Transferring A Payment

If the account on an event's tender and payment are wrong, you can use the Transfer button on the Payment Event page to transfer the payment to another account.

If the account on the payment is wrong - but the tender is correct, you can use the Transfer button on the Payment page to transfer the payment only to another account.

Unbalanced Payment Events

The system, by default, distributes the sum of a payment event's tenders to the account that remits the tender with a single payment. After distribution the sum of the tenders equals the sum of the payments when the event is first created. We refer to such an event as being `balanced`.

However, it is possible for an event's tender amount to not equal the sum of the payment allocations (i.e., the event becomes `unbalanced`). How? Well, there are several ways this can happen:

- While the system DEFAULTS the payment amount to be the tender amount, you can override the payment amount and therefore make a previously balanced event `unbalanced`.
- While the system DEFAULTS the payment account to be the tender account, you can add additional accounts / amounts and therefore make a previously balanced event `unbalanced`.
- When you cancel a tender (e.g., because a check bounces), the system cancels ALL payments linked to the tenders payment event. If the payment event has multiple tenders, this will cause the event to become `unbalanced`. To correct this situation, you must add payment allocations to equal the amount of uncanceled tenders.
- If you cancel a payment and forget to add another payment for the same amount, the event becomes `unbalanced`. To correct this situation, you must add another payment (or cancel the tender).
- You may delete a tender from an event while its tender control is `open`. If you delete a tender and don't do anything about the related payments, the event becomes `unbalanced`.
- You may add a tender to an event at any time. If you don't allocate the tender amount to an account, the event becomes `unbalanced`.

> **Fastpath:** Refer to Payment Event Exceptions for more information about how the system reminds you about unbalanced payment events.

Copyright © 2024, Oracle and/or its affiliates.

How And When Payments Appear On Bills

Fastpath: Refer to Financial Transactions Created Between Bills for more information.

Tender Management and Workstation Cashiering

When you add a tender, you must identify its **Tender Source**. For example,

- A specific cash drawer ID is the source of tenders remitted to a cashier.
- The notional lockbox ID is the source of tenders interfaced from a lockbox.
- The remittance processor is the source of tenders interfaced from a remittance processor.

Fastpath: For more information, refer to Setting Up Tender Sources.

A tender source's tenders must be balanced against an expected total before they can be deposited at a bank. This periodic balancing requires all tenders to exist in respect of a **Tender Control**. Over time, a tender source may have many tender controls (one per balancing event).

An example of a cashier's cash drawer will help clarify the tender control concept:

- When a cashier starts in the morning, s/he starts with a fresh cash drawer (i.e., one without tenders). Whenever a drawer starts afresh, a new Tender Control must be created because you balance the contents of a drawer.
- A cash drawer typically contains funds to make change. These funds are the tender control's **Opening Balance**.

Note: A tender control's starting balance defaults from its tender source.

- During the day, customers remit tenders to the cashier. Every tender put into the drawer is associated with the drawer's tender control created at the start of the day.
- A tender control's balance increases during the day as tenders are recorded. A cashier can view the balance at any time.
- The cashier can turn in funds to the head cashier during the day. Each turn in event can be recorded in the system. Note well, if the amount of funds in a tender control exceeds the maximum balance defined on the tender control's tender source, a warning is issued to the cashier to remind him/her to turn in funds.
- At some point, the contents of the drawer must be balanced against the total tenders linked to the tender control. When balancing starts, no additional tenders may be put into the tender control. If the cashier receives additional tenders after balancing starts, a new tender control must be created (and the above process starts afresh).
- During the balancing process, some modifications may be made to the tenders associated with the tender control, but no additional tenders may be added. When the tender control is balanced, neither it nor its tenders may be modified.

Fastpath: For more information, refer to Managing Your Cash Drawers.

While the above explanation is true, it isn't complete. In addition to the requirement that a tender must reference a tender control, the tender control must refer to a **Deposit Control**. Deposit controls give you administrative control over all of the tender controls whose contents will be deposited en masse. The following concepts will help explain the power of deposit controls:

- As explained above, when a cash drawer is started afresh, a new tender control must be created. The tender control "holds" all new tenders received by the cashier.
- Similarly, when a tender control is created, it must reference a deposit control. During the day, a deposit control's tender controls are constantly changing. You can view the total impact of a deposit control's tender controls at any time.
- At some point, you will want to deposit the tenders received during the day. To do this, you must indicate how much will be deposited at the bank. This deposit amount must equal the sum of the tender controls linked to the cash drawer. When this deposit balancing starts, no additional tender controls may be associated with the deposit control.

Copyright © 2024, Oracle and/or its affiliates.

- When the deposit total equals the sum of the tender controls, the deposit control becomes `balanced` and no changes may be made to it, its tender controls, or its tender controls' tenders.

Fastpath: For more information, refer to Managing Your Cash Drawers.

Note: Background processes use the same concepts. Tenders that are interfaced from external sources (e.g., lockboxes and remittance processors) make use of the concepts describe above. For example, tenders interfaced from a remittance processor are linked to a tender control and this tender control is linked to a deposit control. The main difference is that the background processes require no human intervention; the system automatically creates tender and deposit controls and sets their states to `balanced` when the interface concludes successfully.

Note: The ACH activation process also creates tender and deposit controls. Refer to Activating Automatic Payments for more information.

The topics in this section elaborate on the tender management concepts described above.

Managing Your Cash Drawers

CAUTION: This section assumes you are familiar with the concepts described in The Lifecycle Of A Deposit Control and The Lifecycle Of A Tender Control.

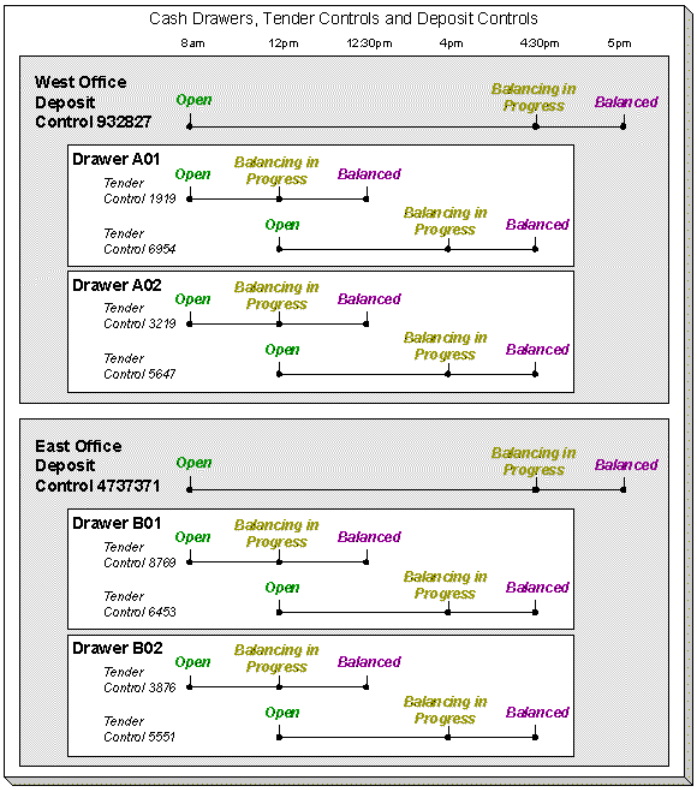
There are many ways to handle the daily management of tenders received via cash drawers. It really depends on how your organization works. To help you understand the potential of the system, we'll continue the example started above.

Assume that the cash drawers in your western office are balanced and deposited independently from those in your eastern office. We'll assume that both offices follows the same daily routine:

- Load fresh drawers first thing in the morning. Each drawer contains a starting balance of $150.00. Note: the drawer's tender control's starting balance defaults from its tender source.
- At 10 am, the cashier turns in funds to the chief cashier and continues to receive additional tenders.
- At 12 noon, each drawer is pulled and balanced by a supervisor.
- By 12:30 pm, the tender controls are balanced.
- At 4 pm, the cashiering stations are closed. Each drawer is pulled and balanced by a supervisor.
- By 4:30 pm, the tender controls are balanced.
- At 5 pm, the deposit control is balanced and funds are ready to be deposited at the bank.

Given this, the following diagram illustrates the deposit controls and tender controls used by each office on a given day.

Copyright © 2024, Oracle and/or its affiliates.



The following concepts are illustrated above:

- An `Open` deposit control must exist before you can create a tender control. And an `Open` tender control must exist before you can create a tender. From a business process standpoint, this means:
  - A supervisor would create a deposit control at the start of the day (8 am in the above illustration).
  - Each cashier would create a tender control when they start a drawer and reference the deposit control created by the supervisor.
  - During the day, the cashier can turn-in moneys to the chief cashier. These turn-in events are recorded in the system as they play a part in the ultimate balancing of the drawer. Refer to Turn Ins for more information.
  - At some point, the contents of a drawer can be pulled and balanced. If additional tenders can be received in a drawer, a new tender control must be created for the drawer. Refer to Balancing By Tender Type for more information.
  - At the end of the day, the supervisor checks to make certain that all tender controls linked to the deposit control are `Balanced`. After this has been done, the supervisor indicates the deposit amount on the deposit control and changes it to `Balanced`. Notice that in the above illustration each deposit control references four tender controls.
- Typically, a cash drawer has one tender control `Open` at any point in time (meaning that the tenders being received are being linked to a specific tender control). However, this is not a hard rule. If you want, you may have multiple tender controls `Open` at any point for a specific cash drawer (for example, if multiple cashiers can work the same drawer during the day but take their drawer with them).
- Typically, a specific cashier puts tenders into a specific tender control. However, this is not a hard rule. On a tender control, you can define if it's limited to a specific operator OR if any operator can link tenders to it.
- When you're ready to balance a drawer, you change the tender control to `Balancing in Progress`. This prevents new tenders from being added to the tender control. If the cashier can continue to receive tenders, s/he must create another tender control. In the above example, all drawers are balanced at 12 noon by a supervisor while the cashier continues to take payments.
- When the tender control is balanced, you change its state to `Balanced`. This prevents any changes to the tender control or its tenders.
- All tender controls exist in respect of a deposit control (in fact, the deposit control must be created before the tender control). This way, a supervisor can check the state of the related drawers throughout the day. Notice that the state

Copyright © 2024, Oracle and/or its affiliates.

transition of a deposit control is identical to that of the tender control (refer to The Lifecycle Of A Deposit Control and The Lifecycle Of A Tender Control). There is only a temporal difference. Notice that the deposit control stays open throughout the day while any number of tender controls are being opened and balanced.

Note: Multiple deposits in a day. While the above example illustrates a single deposit per office per day, it is quite possible to have multiple deposit controls on any given day.

Note: Turns ins. The above example did not illustrate the fact that a cashier can turn-in moneys during the day without having to balance the drawer. Refer to Turn Ins for more information.

Turn Ins

A cashier may optionally turn-in funds received into a cash drawer to a head cashier. The turn-in process requires two steps:

- Each time a cashier turns-in funds, they add a turn-in event on the Tender Control - Turn Ins page. Note that a separate turn-in event is required for each type of tender that's turned in. This is because the balancing of a tender control is performed for each tender type and therefore the system must know how much of each tender type has been removed from the drawer.

Note: Turn in warning. If the amount of cash-like funds in a tender control exceeds the maximum balance defined on the tender control's tender source, a warning is issued to the cashier to remind him/her to turn in funds.

- The head cashier (the person responsible for the deposit control associated with the various tender controls) approves the turn in using the Deposit Control - Turn Ins page.

Note: All turn-ins must be approved for balancing to complete. A tender control cannot be balanced until the head cashier has approved all turn-in events.

Balancing By Tender Type

It should be noted that when it's time to balance a tender control, the cashier must enter the amount of each tender type that is in the drawer in order to balance it. For example, assume the following takes place:

- A drawer is opened with a starting balance of $150.50 (tender type is Cash).
- During the day, the cashier receives $5,000 in Cash, and $1,000 in Checks.
- During the day, the cashier turns in $750 or the Checks and $4,000 of Cash.

At the end of the day, the following operator would have to enter the balances shown in the Ending Balance column.

Tender Type	Starting Balance	Tenders Received	Turn Ins	Ending Balance
Cash	$150.50	$5,000	$4,000	$1150.50
Check	-	$1,000	$750	$250

Note: Time saver. The system assists in the balancing effort by amalgamating the amount of tenders by tender type when the tender control's status is changed from `Open` to `Balancing In Progress`. The cashier just needs to enter the Ending Balance. The system can then compare the cashier's Ending Balance against the Expected Ending Balance. When these values are equal for all tender types, the tender control's status can become `Balanced`.

Cash Back

When a payment is added, the user defines the following:

- The amount of debt to be relieved (i.e., the payment amount)
- The amount remitted (i.e., the tender amount)
- The form of the remittance (i.e., the tender type)

Copyright © 2024, Oracle and/or its affiliates.

The payment amount typically equals the tender amount unless cash will be returned to the customer. For example, if a customer remits $100, but only wants to pay off $25 of debt, the tender amount will be $100 and the payment amount will be $25. The system will only allow a user to remit more than the payment amount if the tender type indicates "cash back" is allowed. For example, you may not allow cash to be returned if a check is remitted, but you may allow it to be returned if cash is remitted.

If cash back is allowed for the tender type, the system displays the amount of cash to be returned on the Payment Event. In addition, because the system enforces balancing the cash drawer by tender type, the system adjusts the payment event's tendered information as follows:

- When there is cash back, the payment event will have two tenders - one will be for the amount and type entered by the user, the other will be a negative amount with a tender type of cash (note, this tender type is retrieved from the Starting Balance Tender Type on the installation record). For example, if a customer remits $100 in traveler's checks, but only wants to pay off $25 of debt, there will be two tenders: one for the $100 travelers check and the other for the -$25 of cash.

Note: Multiple tenders and payment cancellation. If multiple tenders were created because of "cash back" processing, both tenders must be cancelled if the payment event needs to be cancelled.

When modifying an unfrozen payment on the Payment Event, if the payment becomes unbalanced, a button is displayed allowing the user to **Recalculate Cash Back**. If the user clicks this button, the system reassesses the cash back tender as follows:

- If there is now cash back, a new tender is created for the credit amount
- If the cash back amount has changed, the tender for the cash back is adjusted to the new amount
- If there is no longer cash back due, the tender for the cash back is removed.

Managing Payments Interfaced From External Sources

Just like a payment recorded on-line via a cash drawer, a payment interfaced from an external source (e.g., lock box or remittance processor) must reference a tender control and the tender controls, in turn, must reference a deposit control. The only real differences between these two types of payments are highlighted below:

- Whilst an operator must create and balance the tender and deposit controls for real-time payments, the system creates the tender and deposit controls associated with interfaced payments.
- It's impossible for an invalid account to be referenced on a payment recorded real-time. However, it is quite possible for an interfaced payment to reference an unknown account. If an invalid account is referenced on an interfaced payment (using no distribution rules), the system links it to the suspense contract referenced on the tender source control table.

Fastpath: Refer to Interfacing Payments From External Sources for more information.

Exceptions

The topics in this section describe exceptions that are detected when the system allocates a payment.

Payment Exceptions

When the system attempts to distribute a payment, there are a small number of situations where it can't do its job. Some examples of classic errors:

- **No contract to hold a credit.** For example, if an account overpays their debt and the account doesn't have a single contract that is allowed to hold a credit, a payment error is generated.

The system saves payments that are in error just as it saves payments that are error-free. This is done because payments are nothing more than a snapshot of the data that was used to distribute the payment. By saving the snapshot, you can see the information the system used when it detected the error and therefore more effectively correct it.

Copyright © 2024, Oracle and/or its affiliates.

Every payment in error is written to the Payment Exception table. A To Do background process creates To Do entries for records in this table.

Payment Event Exceptions

It is possible for a payment event's tenders to not equal its payments. Such events are classified as `unbalanced`. Refer to Unbalanced Payment Events for how this can happen.

For each unbalanced payment event, a record is written to the Payment Event Exception table. A To Do background process creates To Do entries for records in this table.

Resolving Exceptions Automatically

Some payment errors occur because master data was not fully set up prior to receiving a payment for the account. For these cases, the system will periodically check to see whether the master data problem has been resolved by attempting to distribute and freeze the payment in error.

A background process, Resolve Payments in Error - PY-RPE , exists for this purpose. This background process works as follows:

- It looks for payments in error where the error was caused by the lack of active contracts linked to the payment's account. Note, this may occur when a customer is verbally asked to pay a connection fee or deposit prior to starting service, and for some reason, the appropriate contract has not been set up yet, or is still pending.
- For each such payment, it attempts to re-distribute the payment. If contracts have been created in the meantime, this payment will distribute and freeze successfully.

Payment Financial Transaction Considerations

A payment segment exists for each contract that receives a portion of a payment. Linked to every frozen payment segment is a financial transaction. This financial transaction affects a contract's payoff balance and/or current balance. It also contains the journal details that debt cash and credit some other account.

The topics in this section provide information about the financial impact of a payment segment.

Fastpath: Refer to The Financial Big Picture for more information about a payment's place in the financial big picture.

Payment - Current Balance versus Payoff Balance

CAUTION: If you do not understand the difference between payoff balance and current balance, refer to Current Amount versus Payoff Amount.

A payment segment financial transaction almost always affects payoff balance and current balance by the same amount (think of it like this - when a customer pays, the amount they think they owe goes down by the amount they really owe). The only exception is a payment segment for a charitable contribution. These payment segments only affect current balance because the customer was never billed for the contribution in the first place.

Fastpath: Refer to Setting Up Payment Segment Types for more information about how payment segment type affects how a payment segment is produced and how its financial transaction is generated.

The Source Of GL Accounts On A Payment Financial Transaction

A payment segment's financial transaction also contains the double-sided accounting entry that defines how the payment segment affects the general ledger.

Fastpath: Refer to The Source Of GL Accounts On Financial Transactions for a description of where the system extracts the distribution codes used to construct the GL accounts.

Copyright © 2024, Oracle and/or its affiliates.

A Payment May Affect More Than Just Customer Balances

The topics in this section provide information about obscure things that may happen when a payment is distributed and frozen.

Open Item Accounting and Match Events

Fastpath: Refer to Payments and Match Events for more information about how payments can create match events for open-item accounts.

FT Freeze Repercussions

Fastpath: Refer to Obscure Things That Can Happen for more information about things that can happen when an FT is frozen (and FT's get frozen when a payment is frozen).

Automatic Payments

This section discusses how to set up and manage customers who pay their bills automatically (via direct debit or credit card debits)

How To Set Up A Customer To Pay Automatically

If a customer wants to pay automatically, transfer to Account - Auto Pay and define the source of the funds and the customer's account or credit card number.

What Are Automatic Payments?

An automatic payment is just like any other payment (refer to A Payment Event Has Payments And Tenders for more information about payments in general). However, automatic payments have one special trait - they cause funds to be transferred into your company's bank account. Refer to Downloading Automatic Payments and Interfacing Them To The GL for how this transference happens.

How And When Are Automatic Payments Created?

The system creates automatic payments for bills which are created for accounts with an active auto pay option (note, the system uses the account's auto pay option that's effective on the bill's due date). The **Autopay Creation Option** field in the **Billing** tab of the **Installation Options** screen controls when automatic payments are created, distributed, and frozen. The valid values are:

- **Create At Bill Completion** - On selecting this auto pay creation option, the automatic payment is created when the account's bill is `completed`. The payment is distributed and frozen at the same time. This means that the account's balance is almost always zero (or in credit if they have a deposit). You will see the automatic payments in the payment list immediately after the bill is completed (rather than waiting until the automatic payment's extraction date).
- **Create On Extract Date** - On selecting this auto pay creation option, the automatic payment is not created when the bill is completed. Instead, the system calculates the automatic payment amount and extract date during the bill completion and stamps these details on the bill. The date is the automatic payment source's extraction date (refer to Automatic Payment Dates for more information on how this date is calculated). You can then create automatic payments for such bills on the extract date through the APAYCRET batch.

Note: If a customer with an account that is set up for automatic payment has a non-billed budget that is not excluded from automatic payment, a background process (NBBAPAY) creates an automatic payment on the scheduled payment dates.

The automatic payment is not distributed and frozen when the automatic payment is initially created. A separate background process (APAYDSFR) distributes and freezes the automatic payment on the automatic payment GL

Copyright © 2024, Oracle and/or its affiliates.

distribution date (refer to *Automatic Payment Dates* for more information on how this date is calculated). This means that the customer's balance increases when the bill is completed and is only reduced when the automatic payment is marked for interface to the general ledger.

> **Note:**
>
> It is possible for automatic payments to be distributed and frozen after being extracted and interfaced to a financial institution. Please refer to Downloading Automatic Payments and Interfacing Them To The GL and The Nightly Processes.
>
> An algorithm plugged in on the Installation Record calculates the payment amount irrespective whether the automatic payment is created at bill completion time or on the extract date. Please refer to APAM-DFLT for more information about how the algorithm that is supplied with the base package calculates this amount.
>
> **With balance forward accounting, automatic payments are not just for new charges.** The base package algorithm includes prior balances when it creates a customer's first automatic payment. For example, if a customer has an existing balance of $100 and then signs up for automatic payment, their next bill will cause an automatic payment of $100 plus any new charges to be created (assuming the $100 remains unpaid at the time the next bill is completed). Refer to Open Item Versus Balance Forward Accounting for information about balance forward accounting.

- If a customer with a promise to pay indicates that the payment method is via "autopay", a background process (PPAPAY) creates an automatic payment on the scheduled payment dates. Please note, if the **Autopay Creation Option** on the installation record is set to `Create On Extract Date`, the automatic payment is NOT distributed and frozen when the automatic payment is initially created. Rather, a separate background process (APAYDSFR) distributes and freezes the automatic payment on the automatic payment GL distribution date (refer to *Automatic Payment Dates* for more information on how this date is calculated). Refer to The Big Picture Of Promise To Pay for more information about promise to pay.
- If a customer with an account that is set up for automatic payment has a non-billed budget that is not excluded from automatic payment, a background process (NBBAPAY) creates an automatic payment on the scheduled payment dates. Please note, if the **Autopay Creation Option** on the installation record is set to `Create On Extract Date`, the automatic payment is NOT distributed and frozen when the automatic payment is initially created. Rather, a separate background process (APAYDSFR) distributes and freezes the automatic payment on the automatic payment GL distribution date (refer to *Automatic Payment Dates* for more information on how this date is calculated). Refer to *What is a Non-billed Budget* for more information about non-billed budgets.
- A user can create an automatic payment by simply adding a payment tender with a tender type that indicates it is for automatic payment purposes. This would be a rather unusual thing to do, but you might do this if you want to immediately debit a customer's bank account after a large adjustment is added to the system (e.g., if they suddenly owe you a lot of money and you don't want to wait until the next bill to collect it). Automatic payments created by this method must be distributed and frozen before they can be extracted.

> **Note: An algorithm is used to create automatic payments.** The logic used to create automatic payments is plugged in on the Installation Record. Please note that this algorithm is **not** called when a user manually creates an automatic payment (by adding a payment tender with a tender type that indicates that it is for automatic payment purposes).

When an automatic payment is first created, it gets marked with a distribution date. The distribution date is the date on which the automatic payment's FT's GL details can be interfaced to the general ledger (via the standard GL interface). The distribution date is determined as follows:

- Every automatic payment references an auto-pay source.
- Every auto-pay source references an auto-pay route type.
- Every auto-pay route type contains an algorithm that is responsible for calculating the GL Distribution (Posting) date. On the GL distribution date, the automatic payment will be interfaced to the general ledger.

- **Freeze Payment on Notification** - On selecting this auto pay creation option, the automatic payment is created and distributed when the account's bill is `completed`. However, the automatic payment is frozen or cancelled for the account once the review comments are received from the auto clearing house. For more information, refer to the Freeze Payments on Notification section.

Copyright © 2024, Oracle and/or its affiliates.

The system allows you to override the **Create On Extract Date** and **Freeze Payment on Notification** options for a set of accounts so that automatic payments are frozen immediately during the bill completion. For more information, refer to the Override Automatic Payment Creation Option section.

Note: An algorithm is used to create automatic payments. The logic used to create automatic payments is plugged in on the Installation Record. Please note that this algorithm is **not** called when a user manually creates an automatic payment (by adding a payment tender with a tender type that indicates that it is for automatic payment purposes).

Automatic Payment Dates

As described in the previous section, an algorithm (that's plugged in on Auto Pay Route Types) controls the date on which the automatic payment is interfaced to your general ledger. We refer to this date as the GL distribution date.

This algorithm also populates the following dates:

- The payment date that is stored on the payment.
- The date on which the automatic payment is interfaced to the financial institution.

The algorithm that is supplied with the base package provides many parameters that allow you to dictate how these dates are calculated. Please refer to APAY-DTCALC for the details.

How To Implement Maximum Withdrawal Limits

In some locales, customers can define a "maximum withdrawal amount" to limit the amount of money that is automatically debited from their bank account. For example, a low-income customer may want to prevent direct debits of more than $50 from being applied to their checking account.

You define a customer's maximum withdrawal amount when you setup their automatic payment information on Account - Auto Pay.

The following points describe how the system implements maximum withdrawal limits:

- When a bill is completed for a customer who pays automatically the system calls the automatic payment creation algorithm that's plugged in on the Installation Record. The base-package autopay creation algorithm checks if the amount of the automatic payment exceeds the customer's maximum withdrawal amount. If so, the autopay creation algorithm calls the `automatic payment over limit algorithm` that's plugged in on the account's customer class. Algorithms of this type have the ability to reduce the amount of the autopay or to prevent the autopay from being created. Refer to APOL-RA for an example plug-in.
- When a user manually creates an automatic payment (by adding a tender with a tender type that indicates that it is for automatic payment purposes), the system issues a warning message when the tender amount exceeds the account's maximum withdrawal amount.

Note: Promise To Pay and Non-billed budgets. Please note that automatic payments that are created as a result of promise to pay and non-billed budgets are not subject to maximum withdrawal limits. This is because both of these options required customer approval and therefore the customer should be able to plan accordingly. Refer to How And When Are Automatic Payments Created for more information.

How Are Automatic Payments Cancelled?

There are two ways to cancel an automatic payment:

- The system will cancel an automatic payment behind-the-scenes if the related bill (if any) is reopened BEFORE the automatic payment is interfaced to the financial institution. When you recomplete the bill, the system will create a new automatic payment that reflects the new amount due (and the canceled automatic payment will net out the original automatic payment).
- An operator can cancel an automatic payment (refer to How To Cancel A Tender) at any time. You would do this if the automatic payment was rejected by the financial institution.

Refer to Credit / Correction Notes and Auto Pay for information about how a credit note may cancel an automatic payment.

Copyright © 2024, Oracle and/or its affiliates.

Match Events Are Created For Open-Item Customers When An Automatic Payment Is Created

The system creates a match event when a bill is completed for open-item customers that pay automatically (i.e., direct debit customers). The match event groups together the bill's new charges against the automatic payment's payment segments.

If the bill is subsequently re-opened, the match event will be cancelled when the automatic payment is cancelled.

Fastpath: Refer to Open Item Accounting for more information.

Promise To Pay and Automatic Payments

If a customer wants to pay their promise to pay scheduled payments automatically, the account must be set up for automatic payment (as described under How To Set Up A Customer To Pay Automatically). In addition, the payment method defined on the promise to pay must indicate automatic payment is being used.

When this is done, a background process referred to as `PPAPAY` creates automatic payments on the scheduled payment date by calling the automatic payment creation algorithm plugged in on the installation record.

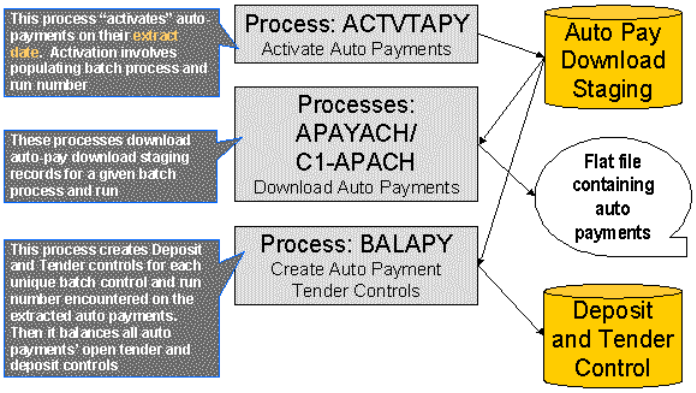
Non-billed Budgets and Automatic Payment

If a customer wants to pay their non-billed budget scheduled payments automatically, the account must be set up for automatic payment (as described under How To Set Up A Customer To Pay Automatically). In addition, the non-billed budget must indicate that automatic payment is being used.

When this is done, a background process referred to as `NBBAPAY` creates automatic payments on the scheduled payment date by calling the automatic payment creation algorithm plugged in on the installation record.

Downloading Automatic Payments and Interfacing Them To The GL

The following diagram illustrates the background processes that interface automatic payment out of the system:



These processes are described in the following topics.

ACTVTAPY - Activating Automatic Payments

When an automatic payment is first created, it gets marked with an extract date. The extract date is the date the automatic payment will be downloaded to the respective financial institution. The extract date is determined as follows:

- Every automatic payment references an auto-pay source.
- Every auto-pay source references an auto-pay route type.
- Every auto-pay route type contains an algorithm that calculates this date.

On the extract date, the automatic payment is "activated". The automatic payment is activated is marked for download the next time its download process runs. An automatic payment's download process is defined on its auto-pay source's route type.

Copyright © 2024, Oracle and/or its affiliates.

APAYACH - Download Automatic Payments To The ACH (automated clearing house)

This process reads all auto pay download staging records marked with a given batch control ID & run number and creates the flat file that's passed to the ACH. Refer to ACH Record Layouts for the details of the record layouts.

Note: This process may be rerun. You can reproduce the flat file at any time. Simply request this job and specify the run number associated with the historic run.

If you require a different flat file format, you must create additional versions of this program. Refer to Setting Up Automatic Payment Extracts for instructions describing how to add another process.

C1-APACH - Download Automatic Payments To The ACH (automated clearing house)

This process is an alternative to using APAYACH. Similar to APAYACH, this process reads all auto pay download staging records marked with a given batch control ID and run number and creates the flat file that's passed to the ACH. Refer to ACH Record Layouts for the details of the record layouts.

The following outlines the differences between APAYACH and C1-APACH:

- C1-APACH populates the Effective Entry Date field on the Company Batch Header Record with a date (the actual draft date) as to when the payment should be withdrawn from the customer's account. APAYACH uses the Business Process Date as the draft date.
- C1-APACH produces a file with multiple occurrences of the Company Batch Header Record differing on the Effective Entry Date field. APAYACH produces a file that has 1 File Header Record, 1 Company Batch Header Record and multiple detail lines.

Note: This process may be rerun. You can reproduce the flat file at any time. Request this job and specify the run number associated with the historic run.

If you require a different flat file format, you must create additional versions of this program. Refer to Setting Up Automatic Payment Extracts for instructions describing how to add another process.

BALAPY - Creating Automatic Payment Tender Controls

This process creates a new tender control (with an associated deposit control) for each unique batch control and run number encountered in the extracted automatic payments (where its payment tender is not yet linked to a tender control). The payment tender of each of these automatic payments is then linked to the corresponding tender control. This process also balances the open tender control records afterwards.

How To Split Automatic Payment

Through split automatic payment you can accept a payment from multiple auto pay sources (i.e banks) with contribution in percentage of payment amount to be set for each auto pay source.

You can create a payment from multiple auto pay sources with contribution of payment in percentage of auto pay amount. Contribution from each auto pay sources (i.e Bank Accounts) are set as percentage in Auto Pay records.

The payment methods for auto payments are as follows:

- **Bank Account** as payment method.
- **Credit Card** as payment method.

For example customer can setup two Auto Pay Sources under split as follows:

Account ID	Auto Pay ID	Auto Source Code	Auto Pay Route Type	Priority	Percentage	Start Date	End Date
9090908298	12344	Bank of America	ACH	1	60	01-01-2016	
9090908298	67467	Chase Bank	ACH	1	40	01-01-2016	

Copyright © 2024, Oracle and/or its affiliates.

Account ID	Auto Pay ID	Auto Source Code	Auto Pay Route Type	Priority	Percentage	Start Date	End Date
9090908298	74674	Citi Bank	ACH	2	70	01-01-2016	
9090908298	74674	Barclays Bank	ACH	2	30	01-01-2016	

In the above example, set of banks (i.e. Bank of America, Chase Bank) with **Priority** 1 are always eligible for Auto Payment unless they are ended with end dates. Second set of banks are eligible beyond end dates of the 1st set.

- **Bill-1:** $100; Bill Due Date, 1-2-2016 ; Eligible Auto Pay Sources of priority - 1
- **Payment Event-1:** $60; Pay Tender Auto Pay with Auto Pay Source Type=Bank of America; Pay Tender ID=1; Tender Amount $60
- **Payment Event-2:** $40; Pay Tender Auto Pay with Auto Pay Source Type=Chase Bank ; Pay Tender ID=2 ;Tender Amount $40

Account ID	Auto Pay ID	Auto Source Code	Auto Pay Route Type	Priority	Percentage	Start Date	End Date
9090908298	12344	Bank of America	ACH	1	60	01-01-2016	7-1-2016
9090908298	67467	Chase Bank	ACH	1	40	01-01-2016	7-1-2016
9090908298	74674	Citi Bank	ACH	2	70	01-01-2017	
9090908298	74674	Barclays Bank	ACH	2	30	01-01-2017	

- After 7-1-2016 the set of banks with **Priority** 2 are eligible to source the Auto Payment.

Note:

If splitting of a payment is not required then you need not set any amount in the percentage field, system will automatically convert the contribution from the auto pay source to 100%.

If two or more Auto Pay records exist for the same period then the record with highest priority will be used for auto payment.

Auto-Pay for bill in credit is not supported.

A user can set any combination of **Auto Pay Source Types** and **Auto Pay Route Types**.

Rule Based Auto Pay Instructions

You can define rule based auto pay instructions for an account. If a financial transaction created against a bill satisfies the rules defined in an auto pay instruction, the automatic payment is created using the respective auto pay ID. For example, Bill 1 has three bill segments - BS1, BS2, and BS3. The BS1 and BS3 financial transactions satisfy the rules defined in the auto pay ID 123456. And, the BS2 financial transaction satisfies the rules defined in the auto pay ID 456789. Then, the system will create two automatic payments against a bill - one using the auto pay ID 123456 and another using the auto pay ID 456789. In this way, the system enables you to create multiple automatic payments for a bill using different auto pay ID.

A new screen named **Auto Pay Instruction** enables you to define rule based auto pay instructions for an account. Note that you can define rule based auto pay instructions for an account when the **Rule Based Auto Pay** check box in the **Auto Pay** tab of the **Account** screen is selected.

Copyright © 2024, Oracle and/or its affiliates.

If you want to define a rule based auto pay instruction, you must set the auto pay type to **Regular**. A rule based auto pay instruction is driven through a rule type. While defining rules in an auto pay instruction, you can only use those fields which are selected as the input and output parameters in the specified rule type. Similarly, you can only use characteristics of those entities which are selected as the rule criteria characteristic entities in the specified rule type.

While freezing a financial transaction where the financial transaction type is set to Bill Segment (BS), Bill Cancellation (BX), Adjustment (AD), and Adjustment Cancellation (AX), the system checks whether there are rule based auto pay instructions for the account which are effective on the bill's due date. If there are multiple rule based auto pay instructions effective on the bill's due date, the system will sort and consider the rule based auto pay instructions based on the priority. If the rule based auto pay instruction with the highest priority does not have any rules, the auto pay ID is simply stamped against the financial transaction in the **C1_FT_EXT** table. If the rule based auto pay instruction with the highest priority has rules, then the rule criteria derivation algorithm on the respective rule type is executed to derive the following from the financial transaction:

- Values of all those fields (such as, policy number, plan number, or price item) which are selected as the input and output parameters in the rule type
- Characteristics of all those entities which are selected as the rule criteria characteristic entities in the rule type

Note: The characteristics which are effective at the time of FT freeze or bill completion are derived depending on the value defined for the **Bill Segment Freeze Option** field in the installation options.

The output is then used by the auto pay rules engine to check whether the financial transaction satisfies any rule defined in the auto pay instruction. If the financial transaction satisfies any rule defined in the auto pay instruction, the auto pay ID is stamped against the financial transaction in the **C1_FT_EXT** table. However, if the financial transaction does not satisfy any rule defined in the auto pay instruction, the system considers the rule based auto pay instruction with the next priority and then derives the parameters which are to be used by the auto pay rules engine for evaluation. If the financial transaction does not satisfy any rule based auto pay instruction which is effective on the bill's due date, the auto pay ID is not stamped against the financial transaction in the **C1_FT_EXT** table.

The following table explains how to enable the rule based auto pay feature during various system events:

System Event	Then...
FT Freeze	You need to define an algorithm using the **C1-APRULEEVL** algorithm type and attach it to the **FT Freeze** system event of the required customer class.
	Note: This algorithm is invoked during: • Online billing when the **Bill Segment Freeze Option** field in the installation options is set to **Freeze At Bill Completion** • Batch billing when the **Freeze and Complete** option in the bill cycle is selected
Bill Completion	You need to define an algorithm using the **C1-RULEAPPRC** algorithm type and attach it to the **Rule Based Auto Payment Processing** system event of the required customer class.

Copyright © 2024, Oracle and/or its affiliates.

System Event	Then...
	Note: This algorithm is invoked during: • Online billing when the **Bill Segment Freeze Option** field in the installation options is set to **Freeze At Bill Completion** (Note that, in this case, the algorithm only considers the financial transactions of the post-processing bill segments.) • Online billing when the **Bill Segment Freeze Option** field in the installation options is set to **Freeze At Will**

In addition, you need to do the following to enable the rule based auto pay feature:

- Define a match type where an algorithm created using the **C1-PYBLAPAY** algorithm type is attached to the **Pay Dist Override Algorithm** system event
- Specify the match type in the **Open-Item Account Match Type Code To Be Used For Rule Based Auto Payments** parameter of an algorithm which is created using the **APAY-CREATE** algorithm type

  Note: This match type is then used while creating automatic payments for an account for which rule based auto pay instructions are defined.
- Set the **Allow Caching Of Rule Based Autopay List** option type of the **C1_FINTRANOP** feature configuration.

ACH Record Layouts

The topics in this section describe the layout of the records created by *APAYACH - Download Automatic Payments To The ACH (automated clearing house)* and C1-APACH - Download Automatic Payments To The ACH (automated clearing house).

Note: APAYACH and C1-APACH share most of the same fields. The exceptions are documented in the following tables.

File Header Record

The ACH extract flat file must have one record of this type and it must be the first logical record on file.

Field Name	Format	Source/Value/Description
RECORD-TYPE	A1	"1"
PRIORITY-CD	A2	"01"
RESERVED-01	A1	Spaces
IMMEDIATE-DESTINATION	A9	CI_BANK_ACCOUNT.DFI_ID_NUM
COMPANY-ID	A10	CI_BANK_ACCOUNT.ACCOUNT_NBR
FILE-CRE-DT	A6	YYMMDD. Current date
FILE-CRE-TM	A4	HHMM. Current time
FILE-ID-MODIFIER	A1	"A"
RECORD-SIZE-CONST	N3	"094" The "FILLER" at the end of all but the Entry Detail Record serves to bring the total length of each record up to this constant.

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
BLOCKING-FACTOR	N2	"10"
FORMAT-CD	A1	"1"
ORIG-FIN-INST-NAME	A23	CI_BANK_L.DESCR
COMPANY-NAME	A23	CI_BANK_ACCOUNT_L.DESCR
REFERENCE-CD	A8	Spaces

Company Batch Header Record

The ACH extract flat file must have one record of this type and it must be the second logical record on file.

Field Name	Format	Source/Value/Description
RECORD-TYPE	A1	"5"
SERVICE-CLASS-CD	A3	"200"
COMPANY-NAME	A16	CI_BANK_ACCOUNT_L.DESCR
COMPANY-DISCRETIONARY	A20	Spaces
COMPANY-ID	A10	CI_BANK_ACCOUNT.ACCOUNT_NBR
STD-ENTRY-CLASS	A3	"PPD"
CO-ENTRY-DESCR	A10	"PAYMENT"
EFF-ENTRY-DT **This field applies only to C1-APACH.**	A6	CI_APAY_CLR_STG.SCHED_EXTRACT_DT + NBR-DAYS batch parameter if specified greater than 0
COMPANY-DESCR-DT	A6	Business process date
EFF-ENTRY-DT	A6	Business process date
RESERVED-01	A3	Spaces
ORIGINATOR-STAT-CD	A1	"1"
ORIGIN-DFI-ID	A8	CI_BANK_ACCOUNT.DFI_ID_NUM
BATCH-NBR	N7	The batch number of this batch within the file.

Entry Detail Record

The ACH extract flat file must have one record of this type for every direct debit record.

Field Name	Format	Source/Value/Description
RECORD-TYPE	A1	"6"
TRANSACTION-CD	A2	CI_TENDER_TYPE.EXT_TYPE_FLG
TRANSIT-RTG-NBR	A9	CI_APAY_SRC.EXT_SOURCE_ID
DFI-ACCT-NBR	A17	CI_APAY_CLR_STG.EXT_ACCT_ID
AMOUNT	N8.2	CI_PAY_TNDR.TENDER_AMT
INDIVIDUAL-ID	A15	CI_PAY_TNDR.PAYOR_ACCT_ID

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
INDIVIDUAL-NAME	A22	CI_APAY_CLR_STG.ENTITY_NAME
DISCRETIONARY-DATA	A2	Spaces
ADDENDA-REC-IND	A1	"0"
DFI-ID	A8	CI_BANK_ACCOUNT.DFI_ID_NUM
ENTRY-DETAIL-CNT	N7	Ascending counter starting at 1 and incremented by 1 for each entry detail record

Note: External Account ID. The EXT_ACCT_ID field supports up to 50 characters. If the value entered in the field is longer than that supported by the record layout (17 characters), the value will be truncated.

Company Batch Control Record

The ACH extract flat file must have one record of this type and it must follow the Entry Detail records.

Field Name	Format	Source/Value/Description
RECORD-TYPE	A1	"8"
SERVICE-CLASS-CD	A3	"200"
ENTRY-ADDENDA-CNT	N6	The total number of entry detail records in this batch.
ENTRY-HASH	N10	The product of the first 8 digits of the external source id of the autopay source, multiplied by the number of entry detail records in the batch.
TOTAL-DR-DOLLAR-AMT	N10.2	Total tender amounts of the entry detail records.
TOTAL-CR-DOLLAR-AMT	N10.2	Zero.
COMPANY-ID	A10	CI_BANK_ACCOUNT.ACCOUNT_NBR
RESERVED-01	A19	Spaces
RESERVED-02	A6	Spaces
ORIGIN-DFI-ID	A8	CI_BANK_ACCOUNT.DFI_ID_NUM
BATCH-NBR	N7	The batch number of this batch within the file.

File Control Record

The ACH extract flat file must have one record of this type and it must be the last logical record on file.

Field Name	Format	Source/Value/Description
RECORD-TYPE	A1	"9"
BATCH-CNT	N6	The number of batches in this file.
BLOCK-CNT	N6	Calculation of the total number of records in the file / 10.0 + 0.9

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
ENTRY-ADDENDA-CNT	N8	The total number of entry detail records in this file.
ENTRY-HASH	N10	The sum of the entry hash values on all the batch control records in this file.
TOTAL-DR-DOLLAR-AMT	N10.2	Sum of the total debit entry dollar amounts of all the batches in this file.
TOTAL-CR-DOLLAR-AMT	N10.2	Sum of the total credit entry dollar amounts of all the batches in this file.
RESERVED-01	A39	Spaces

Issuing A Payment Advice Instead Of Creating An Automatic Payment

If the system is configured to send the customer a payment advice (instead of initiating an electronic funds transfer) when a bill is completed, the automatic payment records - i.e. payment event, payment, tender and auto pay clearinghouse staging - are not created. Refer to Payment Advices for more information.

How To Set Up A Customer To Receive Payment Advices

Use Account - Auto Pay to capture the customer's bank details and indicate an auto pay method of `Payment Advice`.

Payment Advice Option Is For Bill-Related Automatic Payments Only

Payment advices can be printed for auto pays that result from completed bills.

An auto pay for a promise to pay scheduled payment will not be created if the account's effective auto pay option is set to `Payment Advice`. The Promise To Pay Auto Pay (`PPAPAY`) batch process will log an error in this case.

An auto pay for a non-billed budget scheduled payment will not be created if the account's effective auto pay option is set to `Payment Advice`. The Non-Billed Budget Promise To Pay Auto Pay (`NBBAPAY`) batch process will log an error in this case.

Manually created automatic payments (i.e. auto pays created via payment event UI) are always processed as direct debit.

Partial Payment Transfer

Oracle Revenue Management and Billing provides the ability for partial payment transfer at both the payment event and payment levels. You can do partial payment transfer using either of the following ways:

- Payment Request
- Mass Payment Transfer
- Payment Web Service

Partial Payment Transfer through Payment Request

While creating a payment transfer request from the **Payment Event Summary** screen, the system calculates and displays the maximum transfer amount when you transfer one or more payments or all payments of a payment event. The maximum transfer amount is set either to the sum of all frozen payments in the payment event or to the sum of one or more selected payments depending on whether payment transfer is done at the payment event or payment level. Note that the system considers only the positive payments which are in the **Frozen** status while calculating the maximum transfer amount. All negative payments irrespective of the status and the positive payments which are in the **Canceled** status are not considered while calculating the maximum transfer amount. By default, the transfer amount is set to the maximum transfer amount. If you want to do a partial payment transfer, you can change the transfer amount. Note that

Copyright © 2024, Oracle and/or its affiliates.

if the payment is negative or maximum transfer amount is zero, then the **Transfer Amount** field is non-editable. Also, note that the payment transfer amount cannot be more than the maximum transfer amount.

Note: At a time, you can transfer maximum 20 payments from the **Payment Event Summary** screen at the payment level. Also, ensure that when you do partial payment transfer at the payment event level, the payment event should not have more than 7000 payments.

While creating a partial payment transfer request, the system fetches the list of frozen payments depending on whether payment transfer is done at the payment event or payment level. If you partially transfer a single payment through a payment transfer request, the system does the following on the processing the payment transfer request:

- Cancels the old payment
- Creates two new payments in the **Frozen** status - one with the transfer amount using the specified match type and another with the remaining amount using the match type of the old payment

For example, the P1 payment is created against the contract of the A1 account, as shown in the following table:

Account	Payment Event	Payment Details	Match Type
A1	PE1	P1	Suspense Contract

Now, when you partially transfer $150 of payment using the match type as Bill from the A1 to A2 account, the system will cancel the old payment and create the new payments as shown in the following table:

Account	Payment Event	Payment Details	Match Type
A1	PE1	P1	Suspense Contract
A2	PE2	P2	Bill
A1	PE1	P3	Suspense Contract

If you partially transfer more than one payment or all payments of a payment event through a payment transfer request, the system identifies the payments which are eligible for transfer from the payment list in the following manner:

1. If the payments are created against the contracts, the system considers the following option types of the **C1_CMO** feature configuration to determine the payments which are eligible for transfer from the payment list:
   - Suspense Contract Type
   - On Account Contract Type
   - Excess Credit Contract Type

   For more information about the **C1_CMO** feature configuration, see Setting the C1_CMO Feature Configuration on page 1088.
2. However, if the payments are created against the bills, the system considers the bill date and bill amount to determine the payments which are eligible for transfer from the payment list.

The payments which are eligible for transfer from the payment list are added in the **C1_PAY_DETAILS** table. The system stamps the transfer priority corresponding to eligible payments in the **TRANSFER_PRIORITY** column of the **C1_PAY_DETAILS** table. The system assigns transfer priority to each eligible payment in the following sequence:

1. Considers payments against the contracts which are created using the contract type specified in the **Suspense Contract Type** option type of the **C1_CMO** feature configuration. If there are multiple payments which are created on the suspense contracts, the system assigns the same transfer priority to these payments. For example, if you partially transfer P1 and P2 from the A1 account to the A2 account through a payment transfer request where P1 is created against C1, P2 is created against C2, and C1 and C2 are both suspense contracts, then the system sets the transfer priority of P1 and P2 to 1 in the payment transfer request.
2. Considers payments against the contracts which are created using the contract type specified in the **On Account Contract Type** option type of the **C1_CMO** feature configuration. For example, if you partially transfer P1 and P2 from the A1 account to the A2 account through a payment transfer request where P1 is created against C1, P2 is created against C2, C1 is a suspense contract, and C2 is an on account contract, then the system sets the transfer

Copyright © 2024, Oracle and/or its affiliates.

priority of P1 to 1 and P2 to 2 in the payment transfer request. However, if you partially transfer P3 from the A1 account to the A2 account through a payment transfer request where P3 is created against C3 and C3 is an on account contract, then the system sets the transfer priority of P3 to 1 in the payment transfer request. If there are multiple payments on the on account contracts, the system assigns the same transfer priority to these payments. For example, if you partially transfer P1, P2, and P3 from the A1 account to the A2 account through a payment transfer request where P1 is created against C1, P2 is created against C2, P3 is created against C3, C1 is a suspense contract, and C2 and C3 are both on account contracts, then the system sets the transfer priority of P1 to 1 and P2 and P3 to 2 in the payment transfer request.

3. Considers payments against the contracts which are created using the contract type specified in the **Excess Credit Contract Type** option type of the **C1_CMO** feature configuration. For example, if you partially transfer P1 and P2 from the A1 account to the A2 account through a payment transfer request where P1 is created against C1, P2 is created against C2, C1 is an on account contract, and C2 is an excess credit contract, then the system sets the transfer priority of P1 to 1 and P2 to 2 in the payment transfer request. However, if you partially transfer P3 from the A1 account to the A2 account through a payment transfer request where P3 is created against C3 and C3 is an excess credit contract, then the system sets the transfer priority of P3 to 1 in the payment transfer request. If there are multiple payments on the excess credit contracts, the system assigns the same transfer priority to these payments. For example, if you partially transfer P1, P2, and P3 from the A1 account to the A2 account through a payment transfer request where P1 is created against C1, P2 is created against C2, P3 is created against C3, C1 is a suspense contract, and C2 and C3 are both excess credit contracts, then the system sets the transfer priority of P1 to 1 and P2 and P3 to 2 in the payment transfer request.
4. Considers payments created against the bills in the descending order of the bill date (i.e. bill with the latest date). If there are multiple payments which are created against the bills having the same bill date, the system considers the payments which are created against the bills in the descending order of the bill amount (i.e. bill with the highest amount). However, if there are multiple payments which are created against the bills having the same bill date and bill amount, the system assigns the same transfer priority to these payments. For example, if you partially transfer P1, P2, P3, and P4 from the A1 account to the A2 account through a payment transfer request where P1 is created against B1 (with the bill date 01-Jan-2020 and bill amount $100), P2 is created against B2 (with the bill date 01-Feb-2022 and bill amount $20), P3 is created against B3 (with the bill date 01-Mar-2022 and bill amount $30), and P4 is created against B4 (with the bill date as 01-Mar-2022 and bill amount $40), the system sets the transfer priority of P4 to 1, P3 to 2, P2 to 3, and P1 to 4, respectively, in the payment transfer request.
5. Considers payments which are created against entities other than contracts or bills and assigns the same transfer priority to these payments.

In addition, the system sets the **PY_CANCEL_SW** column corresponding the payments in the **C1_PAY_DETAILS** table to **Y** or **N** depending on whether these eligible payments are considered for transfer or not. Let us assume that the following payments are made to the A1 account through the PE1 payment event:

Payment Details	**Match Type**	**Match Value**	**Contract Type**
P1	Suspense Contract	C1	CT1
P2	On Account Contract	C2	CT2
P3	Suspense Contract	C3	CT3
P4	Overpayment on Bill	C4	CT4
P5	On Account Contract	C5	CT5
P6	Suspense Contract	C1	CT1
P7	On Account Contract	C5	CT5
P8	Overpayment on Bill	C4	CT4
P10	Bill	Bill1	
P11	Bill	Bill2	
P12	Bill	Bill3	

Copyright © 2024, Oracle and/or its affiliates.

Payment Details	Match Type	Match Value	Contract Type
P13	Bill	Bill2	

And, let us assume that the **Suspense Contract Type**, **On Account Contract Type**, and **Excess Credit Contract Type** option types of the **C1_CMO** feature configuration are set to CT1, CT5, and CT4, respectively.

Now, if you want to partially transfer $450 of payment from the A1 account to the A2 account using the match type as Bill, the system considers the following payments for transfer from the payment list using the given transfer priority:

Payment Details	Match Type	Match Value	Contract Type	Bill Date
P1	Suspense Contract	C1	CT1	
P2	On Account Contract	C2	CT2	
P3	Suspense Contract	C3	CT3	
P4	Overpayment on Bill	C4	CT4	
P5	On Account Contract	C5	CT5	
P6	Suspense Contract	C1	CT1	
P7	On Account Contract	C5	CT5	
P8	Overpayment on Bill	C4	CT4	
P10	Bill	Bill1		21-Mar-2021
P11	Bill	Bill2		21-Feb-2021
P12	Bill	Bill3		21-Jan-2021
P13	Bill	Bill2		22-Feb-2021

In above example, the system does not consider the P12 payment for transfer even if it is eligible for transfer simply because the transfer amount is exhausted.

If the remaining transfer amount is greater than zero after considering frozen payments against the above mentioned contracts and bills of the account, the system will randomly consider other payments from the payment list which are created against entities other than the above mentioned contracts and bills of the account and assign the same transfer priority to all such payments.

Once the system identifies the payments eligible for transfer from the payment list, the payment transfer request is created in the **Draft** status. In other words, the **BO_STATUS_CD** column corresponding to the payment transfer request in the **C1_PAY_REQ** table is set to **DRAFT**.

The system enables you to edit the transfer amount in the **Draft** status. You can change the transfer amount using the **Update Transfer Amount** button in the **Record Actions** section of the **Payment Request** screen. On clicking the **Update Transfer Amount** button, the **Update Transfer Amount** screen appears where you can change the transfer amount, if required. Note that the updated transfer amount should not be more than the maximum transfer amount. On changing the transfer amount, the system determines the payments which are eligible for transfer once again from the payment list.

You have an option to set the defer payment processing count while defining a payment request type. It is used to determine whether the payments which are eligible for transfer should be derived immediately at the runtime or in the deferred mode (i.e. when the **C1-PAYTR** batch is executed) from the payment list. You can set this field only when the **Transfer** option is selected while defining a payment request type. If you do not specify any value in the **Defer Payment Processing Count** field, the system, by default, sets it to 25.

While creating a payment transfer request or while updating the payment transfer amount in a payment transfer request, the system checks whether the number of payments in the payment list is less than or equal to the value specified in the **Defer Payment Processing Count** field of the respective payment request type. If the number of payments is

Copyright © 2024, Oracle and/or its affiliates.

less than or equal to the defer payment processing count, the system determines the payments which are eligible for transfer from the payment list immediately at the runtime. However, if the number of payments is greater than the defer payment processing count, the system determines the payments which are eligible for transfer from the payment list in the deferred mode. The status of the payment transfer request is then set to **Payment Derivation Pending**.

On executing the **C1-PAYTR** batch, the system checks whether there are any payment transfer requests in the **Payment Derivation Pending** status. If there are any payment transfer requests in the **Payment Derivation Pending** status, the system identifies the payments which are eligible for transfer from the payment list of the payment transfer request in the following manner:

1. If the payments are created against the contracts, the system considers the following option types of the **C1_CMO** feature configuration to determine the payments which are eligible for transfer from the payment list:
   - Suspense Contract Type
   - On Account Contract Type
   - Excess Credit Contract Type

   For more information about the **C1_CMO** feature configuration, see Setting the C1_CMO Feature Configuration on page 1088.
2. However, if the payments are created against the bills, the system considers the bill date and bill amount to determine the payments which are eligible for transfer from the payment list.

The payments which are eligible for transfer from the payment list are added in the **C1_PAY_DETAILS** table. The system stamps the transfer priority corresponding to eligible payments in the **TRANSFER_PRIORITY** column of the **C1_PAY_DETAILS** table. The system assigns transfer priority to each eligible payment in the above mentioned sequence. Finally, the system changes the status of the payment transfer request from **Payment Derivation Pending** to **Draft**. For more information about the batch, refer to *Oracle Revenue Management and Billing Batch Guide*.

On processing the partial payment transfer request, the system will cancel all payments which are transferred and then create new payments using the match type specified in the payment transfer request. Here, the status of the old payments is changed to **Canceled** and the status of the new payments is set to **Frozen**. In addition, if there is a remaining balance on any payment which is considered for transfer, the system will create a new payment for the remaining amount using the match type which is specified in the old payment. The payment for the remaining amount is created in the **Frozen** status. In the above example, the system will cancel the old payments and create the new payments, as shown in the following table:

Payment Details	**Match Type**	**Match Value**	**Contract Type**	**Bill**
P1	Suspense Contract	C1	CT1	
P4	Overpayment on Bill	C4	CT4	
P5	On Account Contract	C5	CT5	
P6	Suspense Contract	C1	CT1	
P7	On Account Contract	C5	CT5	
P10	Bill	Bill1		21-M
P11	Bill	Bill2		21-F
P13	Bill	Bill2		22-F
P14	Bill	Bill4		21-Ja
P15	Bill	Bill2		21-F

Partial Payment Transfer through Mass Payment Transfer

You can upload the data for partial payment transfer from the CSV file using the **File Upload Interface** utility. The transfer amount is required for partial payment transfer. Therefore, the **Transfer Amount** column should be added in

Copyright © 2024, Oracle and/or its affiliates.

the CSV file while uploading the data for partial payment transfer. You should update the existing file request type to include the **Transfer Amount** field in the **Field Transformation** section.

You can specify the transfer amount either less than or equal to the payment amount or to sum of all frozen payments in the payment event depending on whether payment transfer is done at the payment or payment event level. If you do not specify the transfer amount or if the transfer amount is equal to the payment amount or to the sum of all frozen payments in the payment event, the system does the full payment transfer. However, if the transfer amount is less than the payment amount or the sum of all frozen payments in the payment event, the system does the partial payment transfer. Note that the payment transfer amount should not be more than the payment amount or the sum of all frozen payments in the payment event. Also, note that you should only transfer positive payments which are in the **Frozen** status and where the payment amount is greater than zero using the **File Upload Interface** utility.

While doing partial payment transfer at the payment level, the system cancels the old payment and creates two new payments in the **Frozen** status - one with the transfer amount using the specified match type and another with the remaining amount (if any) using the match type of the old payment. For example, the P1 payment is created against the contract of the A1 account, as shown in the following table:

Account	**Payment Event**	**Payment Details**	**Match Type**	**Match Value**	**Amount**	**Status**
A1	PE1	P1	Suspense Contract	C1	$200	Frozen

Now, when you partially transfer $150 of payment using the match type as Bill from the A1 to A2 account, the system will cancel the old payment and create the new payments as shown in the following table:

Account	**Payment Event**	**Payment Details**	**Match Type**	**Match Value**	**Amount**	**Status**
A1	PE1	P1	Suspense Contract	C1	$200	Canceled
A2	PE2	P2	Bill	Bill1	$150	Frozen
A1	PE1	P3	Suspense Contract	C1	$50	Frozen

While doing partial payment transfer at the payment event level, the system fetches the list of frozen payments linked to the payment event and then identifies the payments which are eligible for transfer from the payment list in the following manner:

1. If the payments are created against the contracts, the system considers the following option types of the **C1_CMO** feature configuration to determine the payments which are eligible for transfer from the payment list:
   - Suspense Contract Type
   - On Account Contract Type
   - Excess Credit Contract Type

   For more information about the **C1_CMO** feature configuration, see Setting the C1_CMO Feature Configuration on page 1088.
2. However, if the payments are created against the bills, the system considers the bill date and bill amount to determine the payments which are eligible for transfer from the payment list.

The payments which are eligible for transfer from the payment list are added in the **C1_PAY_DETAILS** table. The system stamps the transfer priority corresponding to eligible payments in the **TRANSFER_PRIORITY** column of the **C1_PAY_DETAILS** table. The system assigns transfer priority to each eligible payment in the following sequence:

1. Considers payments against the contracts which are created using the contract type specified in the **Suspense Contract Type** option type of the **C1_CMO** feature configuration. If there are multiple payments which are created on the suspense contracts, the system assigns the same transfer priority to these payments. For example, if you partially transfer P1 and P2 from the A1 account to the A2 account through a payment transfer request where P1 is

Copyright © 2024, Oracle and/or its affiliates.

created against C1, P2 is created against C2, and C1 and C2 are both suspense contracts, then the system sets the transfer priority of P1 and P2 to 1 in the payment transfer request.

2. Considers payments against the contracts which are created using the contract type specified in the **On Account Contract Type** option type of the **C1_CMO** feature configuration. For example, if you partially transfer P1 and P2 from the A1 account to the A2 account through a payment transfer request where P1 is created against C1, P2 is created against C2, C1 is a suspense contract, and C2 is an on account contract, then the system sets the transfer priority of P1 to 1 and P2 to 2 in the payment transfer request. However, if you partially transfer P3 from the A1 account to the A2 account through a payment transfer request where P3 is created against C3 and C3 is an on account contract, then the system sets the transfer priority of P3 to 1 in the payment transfer request. If there are multiple payments on the on account contracts, the system assigns the same transfer priority to these payments. For example, if you partially transfer P1, P2, and P3 from the A1 account to the A2 account through a payment transfer request where P1 is created against C1, P2 is created against C2, P3 is created against C3, C1 is a suspense contract, and C2 and C3 are both on account contracts, then the system sets the transfer priority of P1 to 1 and P2 and P3 to 2 in the payment transfer request.
3. Considers payments against the contracts which are created using the contract type specified in the **Excess Credit Contract Type** option type of the **C1_CMO** feature configuration. For example, if you partially transfer P1 and P2 from the A1 account to the A2 account through a payment transfer request where P1 is created against C1, P2 is created against C2, C1 is an on account contract, and C2 is an excess credit contract, then the system sets the transfer priority of P1 to 1 and P2 to 2 in the payment transfer request. However, if you partially transfer P3 from the A1 account to the A2 account through a payment transfer request where P3 is created against C3 and C3 is an excess credit contract, then the system sets the transfer priority of P3 to 1 in the payment transfer request. If there are multiple payments on the excess credit contracts, the system assigns the same transfer priority to these payments. For example, if you partially transfer P1, P2, and P3 from the A1 account to the A2 account through a payment transfer request where P1 is created against C1, P2 is created against C2, P3 is created against C3, C1 is a suspense contract, and C2 and C3 are both excess credit contracts, then the system sets the transfer priority of P1 to 1 and P2 and P3 to 2 in the payment transfer request.
4. Considers payments created against the bills in the descending order of the bill date (i.e. bill with the latest date). If there are multiple payments which are created against the bills having the same bill date, the system considers the payments which are created against the bills in the descending order of the bill amount (i.e. bill with the highest amount). However, if there are multiple payments which are created against the bills having the same bill date and bill amount, the system assigns the same transfer priority to these payments. For example, if you partially transfer P1, P2, P3, and P4 from the A1 account to the A2 account through a payment transfer request where P1 is created against B1 (with the bill date 01-Jan-2020 and bill amount $100), P2 is created against B2 (with the bill date 01-Feb-2022 and bill amount $20), P3 is created against B3 (with the bill date 01-Mar-2022 and bill amount $30), and P4 is created against B4 (with the bill date as 01-Mar-2022 and bill amount $40), the system sets the transfer priority of P4 to 1, P3 to 2, P2 to 3, and P1 to 4, respectively, in the payment transfer request.
5. Considers payments which are created against entities other than contracts or bills and assigns the same transfer priority to these payments.

In addition, the system sets the **PY_CANCEL_SW** column corresponding the payments in the **C1_PAY_DETAILS** table to **Y** or **N**depending on whether these eligible payments are considered for transfer or not. Let us assume that the following payments are made to the A1 account through the PE1 payment event:

Payment Details	**Match Type**	**Match Value**	**Contract Type**	**Bill Date**	**Amount**	**Status**
P1	Suspense Contract	C1	CT1		$100	Frozen
P2	On Account Contract	C2	CT2		$50	Frozen
P3	Suspense Contract	C3	CT3		$100	Frozen
P4	Overpayment on Bill	C4	CT4		$100	Frozen

Copyright © 2024, Oracle and/or its affiliates.

Payment Details	Match Type	Match Value	Contract Type	Bill Date	Amount	Status
P5	On Account Contract	C5	CT5		$100	Frozen
P6	Suspense Contract	C1	CT1		$100	Frozen
P7	On Account Contract	C5	CT5		$100	Frozen
P8	Overpayment on Bill	C4	CT4		$75	Canceled
P10	Bill	Bill1		21-Mar-2021	$200	Frozen
P11	Bill	Bill2		21-Feb-2021	$100	Frozen
P12	Bill	Bill3		21-Jan-2021	$350	Frozen
P13	Bill	Bill2		22-Feb-2021	$25	Frozen

And, let us assume that the **Suspense Contract Type**, **On Account Contract Type**, and **Excess Credit Contract Type** option types of the **C1_CMO** feature configuration are set to CT1, CT5, and CT4, respectively.

Now, if you want to partially transfer $900 of payment from the A1 account to the A2 account using the match type as Bill, the system considers the following payments for transfer from the payment list using the given transfer priority:

Payment Details	Match Type	Match Value	Contract Type	Bill Date	Amount	Eligible for Transfer	Transfer Priority	Cancel Switch
P1	Suspense Contract	C1	CT1		$100	Y	1	Y
P2	On Account Contract	C2	CT2		$50	N		N
P3	Suspense Contract	C3	CT3		$100	N		N
P4	Overpayment on Bill	C4	CT4		$100	Y	3	Y
P5	On Account Contract	C5	CT5		$100	Y	2	Y
P6	Suspense Contract	C1	CT1		$100	Y	1	Y
P7	On Account Contract	C5	CT5		$100	Y	2	Y
P8	Overpayment on Bill	C4	CT4		$75	N		N
P10	Bill	Bill1		21-Mar-2021	$200	Y	4	Y
P11	Bill	Bill2		21-Feb-2021	$100	Y	6	Y
P12	Bill	Bill3		21-Jan-2021	$350	Y	7	Y
P13	Bill	Bill2		22-Feb-2021	$25	Y	5	Y

Copyright © 2024, Oracle and/or its affiliates.

If the remaining transfer amount is greater than zero after considering frozen payments against the above mentioned contracts and bills of the account, the system will randomly consider other payments from the payment list which are created against entities other than the above mentioned contracts and bills of the account and assign the same transfer priority to all such payments.

While partially transferring at the payment event level, the system will cancel all payments which are transferred and then create new payments using the specified match type. Here, the status of the old payments is changed to **Canceled** and the status of the new payments is set to **Frozen**. In addition, if there is a remaining balance on any payment which is considered for transfer, the system will create a new payment for the remaining amount using the match type which is specified in the old payment. The payment for the remaining amount is created in the **Frozen** status. In the above example, the system will cancel the old payments and create the new payments, as shown in the following table:

Payment Details	Match Type	Match Value	Contract Type	Bill Date	Amount	Status
P1	Suspense Contract	C1	CT1		$100	Canceled
P4	Overpayment on Bill	C4	CT4		$100	Canceled
P5	On Account Contract	C5	CT5		$100	Canceled
P6	Suspense Contract	C1	CT1		$100	Canceled
P7	On Account Contract	C5	CT5		$100	Canceled
P10	Bill	Bill1		21-Mar-2021	$200	Canceled
P11	Bill	Bill2		21-Feb-2021	$100	Canceled
P12	Bill	Bill3		21-Jan-2021	$350	Canceled
P13	Bill	Bill2		22-Feb-2021	$25	Canceled
P14	Bill	Bill4		21-Jan-2021	$900	Frozen
P15	Bill	Bill3		21-Jan-2021	$275	Frozen

Partial Payment Transfer through Inbound Web Service

You can partially transfer an individual payment or all frozen payments of a payment event using an inbound web service. The transfer amount is required for partial payment transfer. Therefore, the value should be provided for the **transferAmount** tag while doing partial payment transfer through a web service.

You can specify the transfer amount either less than or equal to the payment amount or to sum of all frozen payments in the payment event depending on whether payment transfer is done at the payment or payment event level. If you do not specify the transfer amount or if the transfer amount is equal to the payment amount or to the sum of all frozen payments in the payment event, the system does the full payment transfer. However, if the transfer amount is less than the payment amount or the sum of all frozen payments in the payment event, the system does the partial payment transfer. Note that the payment transfer amount should not be more than the payment amount or the sum of all frozen payments in the payment event. Also, note that you should only transfer positive payments which are in the **Frozen** status and where the payment amount is greater than zero using an inbound web service.

While doing partial payment transfer at the payment level, the system cancels the old payment and creates two new payments in the **Frozen** status - one with the transfer amount using the specified match type and another with the remaining amount (if any) using the match type of the old payment. For example, the P1 payment is created against the contract of the A1 account, as shown in the following table:

Account	Payment Event	Payment Details	Match Type	Match Value
A1	PE1	P1	On Account Contract	C1

Copyright © 2024, Oracle and/or its affiliates.

Now, when you partially transfer $50 of payment using the match type as Bill from the A1 to A2 account, the system will cancel the old payment and create the new payments as shown in the following table:

Account	Payment Event	Payment Details	Match Type	Match Value
A1	PE1	P1	On Account Contract	C1
A2	PE2	P2	Bill	Bill1
A1	PE1	P3	On Account Contract	C1

While doing partial payment transfer at the payment event level, the system fetches the list of frozen payments linked to the payment event and then identifies the payments which are eligible for transfer from the payment list in the following manner:

1. If the payments are created against the contracts, the system considers the following option types of the **C1_CMO** feature configuration to determine the payments which are eligible for transfer from the payment list:
   - Suspense Contract Type
   - On Account Contract Type
   - Excess Credit Contract Type

   For more information about the **C1_CMO** feature configuration, see Setting the C1_CMO Feature Configuration on page 1088.
2. However, if the payments are created against the bills, the system considers the bill date and bill amount to determine the payments which are eligible for transfer from the payment list.

The payments which are eligible for transfer from the payment list are added in the **C1_PAY_DETAILS** table. The system stamps the transfer priority corresponding to eligible payments in the **TRANSFER_PRIORITY** column of the **C1_PAY_DETAILS** table. The system assigns transfer priority to each eligible payment in the following sequence:

1. Considers payments against the contracts which are created using the contract type specified in the **Suspense Contract Type** option type of the **C1_CMO** feature configuration. If there are multiple payments which are created on the suspense contracts, the system assigns the same transfer priority to these payments. For example, if you partially transfer P1 and P2 from the A1 account to the A2 account through a payment transfer request where P1 is created against C1, P2 is created against C2, and C1 and C2 are both suspense contracts, then the system sets the transfer priority of P1 and P2 to 1 in the payment transfer request.
2. Considers payments against the contracts which are created using the contract type specified in the **On Account Contract Type** option type of the **C1_CMO** feature configuration. For example, if you partially transfer P1 and P2 from the A1 account to the A2 account through a payment transfer request where P1 is created against C1, P2 is created against C2, C1 is a suspense contract, and C2 is an on account contract, then the system sets the transfer priority of P1 to 1 and P2 to 2 in the payment transfer request. However, if you partially transfer P3 from the A1 account to the A2 account through a payment transfer request where P3 is created against C3 and C3 is an on account contract, then the system sets the transfer priority of P3 to 1 in the payment transfer request. If there are multiple payments on the on account contracts, the system assigns the same transfer priority to these payments. For example, if you partially transfer P1, P2, and P3 from the A1 account to the A2 account through a payment transfer request where P1 is created against C1, P2 is created against C2, P3 is created against C3, C1 is a suspense contract, and C2 and C3 are both on account contracts, then the system sets the transfer priority of P1 to 1 and P2 and P3 to 2 in the payment transfer request.
3. Considers payments against the contracts which are created using the contract type specified in the **Excess Credit Contract Type** option type of the **C1_CMO** feature configuration. For example, if you partially transfer P1 and P2 from the A1 account to the A2 account through a payment transfer request where P1 is created against C1, P2 is created against C2, C1 is an on account contract, and C2 is an excess credit contract, then the system sets the transfer priority of P1 to 1 and P2 to 2 in the payment transfer request. However, if you partially transfer P3 from the A1 account to the A2 account through a payment transfer request where P3 is created against C3 and C3 is an excess credit contract, then the system sets the transfer priority of P3 to 1 in the payment transfer request. If there are multiple payments on the excess credit contracts, the system assigns the same transfer priority to these payments. For example, if you partially transfer P1, P2, and P3 from the A1 account to the A2 account through a payment transfer request where P1 is created against C1, P2 is created against C2, P3 is created against C3, C1 is a suspense

Copyright © 2024, Oracle and/or its affiliates.

contract, and C2 and C3 are both excess credit contracts, then the system sets the transfer priority of P1 to 1 and P2 and P3 to 2 in the payment transfer request.

4. Considers payments created against the bills in the descending order of the bill date (i.e. bill with the latest date). If there are multiple payments which are created against the bills having the same bill date, the system considers the payments which are created against the bills in the descending order of the bill amount (i.e. bill with the highest amount). However, if there are multiple payments which are created against the bills having the same bill date and bill amount, the system assigns the same transfer priority to these payments. For example, if you partially transfer P1, P2, P3, and P4 from the A1 account to the A2 account through a payment transfer request where P1 is created against B1 (with the bill date 01-Jan-2020 and bill amount $100), P2 is created against B2 (with the bill date 01-Feb-2022 and bill amount $20), P3 is created against B3 (with the bill date 01-Mar-2022 and bill amount $30), and P4 is created against B4 (with the bill date as 01-Mar-2022 and bill amount $40), the system sets the transfer priority of P4 to 1, P3 to 2, P2 to 3, and P1 to 4, respectively, in the payment transfer request.
5. Considers payments which are created against entities other than contracts or bills and assigns the same transfer priority to these payments.

In addition, the system sets the **PY_CANCEL_SW** column corresponding the payments in the **C1_PAY_DETAILS** table to **Y** or **N** depending on whether these eligible payments are considered for transfer or not. Let us assume that the following payments are made to the A1 account through the PE1 payment event:

Payment Details	Match Type	Match Value	Contract Type	Bill Date
P1	Suspense Contract	C1	CT1	
P2	On Account Contract	C5	CT5	
P3	Suspense Contract	C1	CT1	
P4	Overpayment on Bill	C4	CT4	
P5	On Account Contract	C5	CT5	
P6	Suspense Contract	C1	CT1	
P7	On Account Contract	C5	CT5	
P8	Overpayment on Bill	C4	CT4	
P10	Bill	Bill1		21-Mar-202
P11	Bill	Bill2		21-Feb-2021
P12	Bill	Bill3		21-Jan-2021
P13	Bill	Bill2		22-Feb-2021

And, let us assume that the **Suspense Contract Type**, **On Account Contract Type**, and **Excess Credit Contract Type** option types of the **C1_CMO** feature configuration are set to CT1, CT5, and CT4, respectively.

Now, if you want to partially transfer $1400 of payment from the A1 account to the A2 account using the match type as Bill, the system considers the following payments for transfer from the payment list using the given transfer priority:

Payment Details	Match Type	Match Value	Contract Type	Bill Date
P1	Suspense Contract	C1	CT1	
P2	On Account Contract	C5	CT5	
P3	Suspense Contract	C1	CT1	
P4	Overpayment on Bill	C4	CT4	
P5	On Account Contract	C5	CT5	
P6	Suspense Contract	C1	CT1	

Copyright © 2024, Oracle and/or its affiliates.

Payment Details	Match Type	Match Value	Contract Type	Bill Date
P7	On Account Contract	C5	CT5	
P8	Overpayment on Bill	C4	CT4	
P10	Bill	Bill1		21-Mar-2021
P11	Bill	Bill2		21-Feb-2021
P12	Bill	Bill3		21-Jan-2021
P13	Bill	Bill2		22-Feb-2021

If the remaining transfer amount is greater than zero after considering frozen payments against the above mentioned contracts and bills of the account, the system will randomly consider other payments from the payment list which are created against entities other than the above mentioned contracts and bills of the account and assign the same transfer priority to all such payments.

While partially transferring at the payment event level, the system will cancel all payments which are transferred and then create new payments using the specified match type. Here, the status of the old payments is changed to **Canceled** and the status of the new payments is set to **Frozen**. In addition, if there is a remaining balance on any payment which is considered for transfer, the system will create a new payment for the remaining amount using the match type which is specified in the old payment. The payment for the remaining amount is created in the **Frozen** status. In the above example, the system will cancel the old payments and create the new payments, as shown in the following table:

Payment Details	Match Type	Match Value	Contract Type	Bill Date
P1	Suspense Contract	C1	CT1	
P2	On Account Contract	C5	CT5	
P3	Suspense Contract	C1	CT1	
P4	Overpayment on Bill	C4	CT4	
P5	On Account Contract	C5	CT5	
P6	Suspense Contract	C1	CT1	
P7	On Account Contract	C5	CT5	
P8	Overpayment on Bill	C4	CT4	
P10	Bill	Bill1		21-Mar-2
P11	Bill	Bill2		21-Feb-2
P12	Bill	Bill3		21-Jan-2
P13	Bill	Bill2		22-Feb-2
P14	Bill	Bill4		21-Jan-2

Three-Step PUPL Processing

The **Payment Upload (PUPL)** batch faces significant performance issues when the number of payment segments are huge for a single payment event or payment. To handle this performance issue, the system provides an alternative three-step approach wherein you need to execute the following batches in the specified sequence:

1. **PUPL** - The following new parameters are added in the **PUPL** batch:

Copyright © 2024, Oracle and/or its affiliates.

- **Three Step Batch Processing (Y/N)** - Used to indicate whether you want to use three-step approach for uploading payment records from the staging area. The valid values are **Y** and **N**. If you do not specify the value for this parameter or set the parameter value to **N**, you need to just execute the **PUPL** batch in the traditional manner.

  However, if you set the parameter value to **Y**, you need to execute the **PUPL**, **PUPLDF**, and **PUPLCMPL** batches in the specified sequence. In the three-step approach, the **PUPL** batch creates the deposit control, tender controls, payment events, tenders, and payments in the staging area using the payment records from the pre-staging area.

  The status of the payment is set to **Step in Process**. In addition, the payment distribution algorithm is triggered and the payment distribution records are created in the **CI_PAY_DIST_TEMP** table.
- **Division** - Used when you want to create payments for accounts belonging to a particular division.
- **Bulk Insert Count** - Used to indicate the number of records that you want to insert at once in the **CI_PAY_DIST_TEMP** table.

For more information about the batch, refer to *Oracle Revenue Management and Billing Batch Guide*.

2. **PUPLDF** - This batch considers the payment distribution records from the **CI_PAY_DIST_TEMP** table and accordingly creates the payment segments. In addition, it creates the financial transaction for each payment segment and sets its status to **Frozen**. If any of the payment segment returns an error, the system will not hold the creation of other payment segments in the payment. For more information about the batch, refer to *Oracle Revenue Management and Billing Batch Guide*.
3. **PUPLCMPL** - This batch changes the status of the payment to **Frozen** or **Error** depending on whether its payment segments and financial transactions are successfully created in the system. If all payments in a payment event are successfully created, the status of the corresponding deposit control, tender control, payment tender, and payment staging records is set to **Complete**. If any error occurs while creating payment event for a payment record, the status of the deposit control, tender control, payment tender, and payment staging records is changed to **Error**. In addition, the status of the payment record in the pre-staging area is changed to **Error in Staging**. For more information about the batch, refer to *Oracle Revenue Management and Billing Batch Guide*.

Related Topics

For more information on...	See...
Handling multiple payment records of same payee account through Three-Step PUPL Processing	Handling Multiple Payment Records of the Same Payee Account through Three-Step PUPL Batch on page 328

Handling Multiple Payment Records of the Same Payee Account through Three-Step PUPL Batch

Until now, the **Payment Upload (PUPL)** batch failed when the CSV file had more than one payment record for the same payee account during the three-step batch processing. Now, the system enables you to handle this issue using a field mapping. A new preference category named **Configuration for Match Type - Handling Overpayment** is introduced in the **Field Mapping** screen. To define the attributes for field mapping of the **Configuration for Match Type - Handling Overpayment** preference category, refer to the Defining an Overpayment Handling through a Match Type Preference on page 329

Searching for an Overpayment Handling through a Match Type Preference

Prerequisites

To search for an overpayment handling through a match type preference, you should have:

- Values defined for the **Field Mapping for Match Type** option type in the **C1-PAYFLUPLD** feature configuration.

Procedure

To search for an overpayment handling through a match type preference:

1. Click the **Admin** link in the **Application** toolbar.

Copyright © 2024, Oracle and/or its affiliates.

A list appears.

2. From the **Admin** menu, select **F** and then click **Field Mapping**.
   A sub-menu appears.
3. Click the **Search** option from the **Field Mapping** sub-menu.
   The **Field Mapping** screen appears.
4. Select the **Handling Overpayment through Match Type** option from the **Preference Category** list.
5. Enter the additional search criteria in the **Search Field Mapping** zone.

> **Note:** ORMB search engine supports wildcard search, where you can substitute the percentage (%) symbol as a stand in for any word or letter in a search criteria. You can use the '%' wildcard character in all input fields except the date and ID fields. The '%' wildcard character is suffixed automatically at the end of the partial search criteria. Therefore, you may or may not specify the wildcard character at the end of the partial search criteria. However, you have to prefix the wildcard character manually wherever required.

6. Click **Search**.
   A list of overpayment handling through the match type preferences that meet the search criteria appears in the **Search Results** section.

Defining an Overpayment Handling through a Match Type Preference

Prerequisites

To define an overpayment handling through a match type preference, you should have:

- Value defined for the **Field Mapping for Match Type** option type in the **C1-PAYFLUPLD** feature configuration.
- Values defined in the **FIELD_CAT_FLG**, **FIELD_STATUS_FLG**, and **FIELD_NAME_FLG** lookup field.
- Values defined in the **PAY_TND_STG_ST_FLG** lookup field.

Procedure

To define an overpayment handling through a match type preference:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **F** and then click **Field Mapping**.
   A sub-menu appears.
3. Click the **Add** option from the **Field Mapping** sub-menu.

   The **Select Business Object** screen appears. It contains the following field:

Field Name	Field Description	Mandatory (Yes or No)
Field Mapping Business Object	Used to indicate the business object using which you want to create an overpayment handling through a match type preference. The valid values are: • Field Mapping • Medicare Plan Benefit Package • Membership Status Reason Mapping	Yes

> **Note:** Alternatively, you can access the **Select Business Object** screen by clicking the **Add** button in the **Page Title** area of the **Field Mapping** screen.

4. Select the **Field Mapping** option from the **Field Mapping Business Object** list.
5. Click **OK**.

   The **Field Mapping** screen appears. It contains the following sections:

Copyright © 2024, Oracle and/or its affiliates.

- **Main** - Used to specify basic details of the overpayment handling through a match type preference. It contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Preference	Used to specify the overpayment handling through a match type preference.	Yes
Description	Used to specify the description for the overpayment handling through a match type preference.	Yes
Detailed Description	Used to specify additional information about the overpayment handling through a match type preference.	No
Status	Used to indicate the status of the overpayment handling through a match type preference. The valid values are: • Active • Inactive	Yes

- **Preference Category** - This section contains the following field:

Field Name	Field Description	Mandatory (Yes or No)
Preference Category	Used to indicate the category to which the preference belongs. The valid values are: • Automatic Refund/Write Off • Billing • Binder Payment • Handling Overpayment through a Match Type • Geographic Rating Area • Individual Membership • Medicare • Medicare Plan Benefit • Member Reconciliation • Member Relationship and Subscription Tier • Member Status Reason Preferences • Statement Construct **Note:** You must select **Handling Overpayment through a Match Type** option as the preference category while creating an overpayment handling through a match type.	Yes

- **Preference Settings** - Used to set the attributes in the overpayment handling through a match type preference.

Field Name	Field Description	Mandatory (Yes or No)
Attribute	Used to indicate the attribute which you want to set in the overpayment handling through a match type preference. The valid values are: • **Match Type for On Account Payment (C)** - At present, there is no functionality built using this attribute. It would be used in the future release of ORMB.	Yes (Conditional) **Note:** This field is required while defining the settings for an overpayment handling through a match type preference.

Copyright © 2024, Oracle and/or its affiliates.

<table>
<tr><th>Field Name</th><th>Field Description</th><th>Mandatory (Yes or No)</th></tr>
<tr><td></td><td>

- **Match Type for Default Payment Distribution (D)** - Used to indicate the match types for which the system should behave in the following manner:

<table>
<tr><th>PUPL Batch Run</th><th>Then...</th></tr>
<tr><td>First</td><td>The system does the following:
• Processes the first payment record of each payee account and changes the status of the corresponding payment tender staging records to Pending.
• Does not process the subsequent payment records of each payee account and changes the payment tender staging record status of the subsequent payment records to Ignored.</td></tr>
<tr><td></td><td>Note: Once the three-step PUPL batch cycle is completed, the status of the payment record is changed to Processed or Error in Staging depending on whether it is successfully processed or not. In addition, the status of the corresponding payment tender staging records is changed from Pending to Complete.</td></tr>
</table>

</td><td></td></tr>
<tr><td></td><td>

<table>
<tr><th>PUPL Batch Run</th><th>Then...</th></tr>
<tr><td>Second</td><td>The system does the following:
• Processes the second payment record of each payee account and changes the status of the corresponding payment tender staging records from Ignored to Pending.
• Does not process the subsequent payment records of each payee account and the payment tender staging record status of the subsequent payment records remains as Ignored.</td></tr>
</table>

</td><td></td></tr>
</table>

Copyright © 2024, Oracle and/or its affiliates.

<table>
<tr><th>Field Name</th><th>Field Description</th><th>Mandatory (Yes or No)</th></tr>
<tr><td></td><td>
<table>
<tr><th>PUPL Batch Run</th><th>Then...</th></tr>
<tr><td></td><td>Note: Once the three-step PUPL batch cycle is completed, the status of the payment record is changed to Processed or Error in Staging depending on whether it is successfully processed or not. In addition, the status of the corresponding payment tender staging records is changed from Pending to Complete.</td></tr>
</table>
Note: In each three-step PUPL batch cycle, the system considers and processes one payment record of a payee account at a time and therefore you need to repeat the three-step PUPL batch cycle until all payment records of the payee account are processed.
</td><td></td></tr>
<tr><td></td><td>
• Match Type for Overpayment Distribution (O) - Used to indicate the match types for which the system should behave in the following manner:
<table>
<tr><th>PUPL Batch Run</th><th>Then...</th></tr>
<tr><td>First</td><td>The system does the following:
• Processes the first payment record of each payee account and match value combination and changes the status of the corresponding payment tender staging records to Pending.
• Does not process the subsequent payment records of each payee account and match value combination and changes the payment tender staging record status of the subsequent payment records to Ignored.</td></tr>
</table>
</td><td></td></tr>
</table>

Copyright © 2024, Oracle and/or its affiliates.

<table>
<tr><th>Field Name</th><th>Field Description</th><th>Mandatory (Yes or No)</th></tr>
<tr><td></td><td>
<table>
<tr><th>PUPL Batch Run</th><th>Then...</th></tr>
<tr><td></td><td>Note: Once the three-step PUPL batch cycle is completed, the status of the payment record is changed to Processed or Error in Staging depending on whether it is successfully processed or not. In addition, the status of the corresponding payment tender staging records is changed from Pending to Complete.</td></tr>
</table>
</td><td></td></tr>
<tr><td></td><td>
<table>
<tr><th>PUPL Batch Run</th><th>Then...</th></tr>
<tr><td>Second</td><td>The system does the following:
• Processes the second payment record of each payee account and match value combination and changes the status of the corresponding payment tender staging records from Ignored to Pending.
• Does not process the subsequent payment records of each payee account and match value combination and the payment tender staging record status of the subsequent payment records remains as Ignored.</td></tr>
<tr><td></td><td>Note: Once the three-step PUPL batch cycle is completed, the status of the payment record is changed to Processed or Error in Staging depending on whether it is successfully processed or not. In addition, the status of the corresponding payment tender staging records is changed from Pending to Complete.</td></tr>
</table>
Note: In each three-step PUPL batch cycle, the system considers and processes one payment record of each payee account and match value combination at a time and therefore you need to repeat the three-step PUPL batch cycle until all payment records of each payee account and match value combination are processed.
</td><td></td></tr>
</table>

Copyright © 2024, Oracle and/or its affiliates.

<table>
<tr><th>Field Name</th><th>Field Description</th><th>Mandatory (Yes or No)</th></tr>
<tr><td></td><td>

- **Match Type for Segment Level Overpayment Distribution (R)** - Used to indicate the match types for which the system should behave in the following manner:

<table>
<tr><th>PUPL Batch Run</th><th>Then...</th></tr>
<tr><td rowspan="2">First</td><td>The system does the following:
• Processes all payment records of the same or different payee account with the same or different match value and changes the status of the corresponding payment tender staging records to Pending.</td></tr>
<tr><td>Note: Once the three-step PUPL batch cycle is completed, the status of the payment record is changed to Processed or Error in Staging depending on whether it is successfully processed or not. In addition, the status of the corresponding payment tender staging records is changed from Pending to Complete.</td></tr>
</table>

</td><td></td></tr>
</table>

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: You can specify maximum of 20 match types separated by a comma (,) symbol for the above attributes in the field mapping. Now, while executing the step-three PUPL batch, the system considers the field mapping which is specified in the **Field Mapping for Match Type** option type of the **C1-PAYFLUPLD** feature configuration whenever there are multiple payment records of the payee account in the CSV file. In the following scenarios, the system behaves in the manner as if the match type is set to the **Match Type Default Payment Distribution (D)** attribute: • The value for the **Field Mapping for Match Type** option type is not defined in the **C1-PAYFLUPLD** feature configuration • The value specified for the Field Mapping for Match Type option type in the **C1-PAYFLUPLD** feature configuration is invalid • The match type received in the payment record is not specified against any of the attribute in the referred field mapping To implement this change, a new status named **Ignored** is added in the **PAY_TND_STG_ST_FLG** lookup field.	
Value	Used to specify the attribute value.	Yes (Conditional) **Note:** This field is required while defining the settings for an overpayment handling through a match type preference.
Entity Type	Used to indicate the type of entity for which the attribute is applicable. The valid values are: • Account • Adjustment • Billable Charge • Membership • Payment • Policy **Note:** At present, this data is stored for informational purposes only and is not considered in the business logic. If required, you can add more entity types to the **MAP_ENTITY_FLG** lookup field.	No

Note: While defining an overpayment handling through a match type preference, you must set at least one attribute in the overpayment handling through a match type preference.

Copyright © 2024, Oracle and/or its affiliates.

6. Select the **Handling Overpayment through Match Type** option from the **Preference Category** list.
7. Enter the required details in the **Main** section.
8. Select the required attribute and value from the respective fields.
9. If you want to set more than one attribute in the overpayment handling through a match type preference, click the **Add** (+) icon and then repeat step 8.

> **Note:** However, if you want to remove an attribute from the overpayment handling through a match type preference, click the **Delete** (🗑) icon corresponding to the attribute.

10. Click **Save**.
    The overpayment handling through a match type preference is defined.

Editing an Overpayment Handling through a Match Type Preference

Prerequisites

To edit an overpayment handling through a match type preference, you should have:

- Value defined for the **Field Mapping for Match Type** option type in the **C1-PAYFLUPLD** feature configuration.
- Values defined in the **FIELD_CAT_FLG**, **FIELD_STATUS_FLG**, and **FIELD_NAME_FLG** lookup field.
- Values defined in the **PAY_TND_STG_ST_FLG** lookup field.

Procedure

To edit an overpayment handling through a match type preference:

1. Search for an overpayment handling through a match type preference in the **Field Mapping** screen.
2. In the **Search Results** section, click the **Edit** (✎) icon in the **Edit** column *c*orresponding to the overpayment handling through a match type preference whose details you want to edit.

   The **Field Mapping** screen appears. It contains the following sections:

   - **Main** - Used to specify basic details of the overpayment handling through a match type preference. It contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Preference	Displays the overpayment handling through a match type preference.	Not applicable
Description	Used to specify the description for the overpayment handling through a match type preference.	Yes
Detailed Description	Used to specify additional information about the preference.	No
Status	Used to indicate the status of the overpayment handling through a match type preference. The valid values are: • Active • Inactive	Yes
Preference Category	Indicates the feature for which the preference is designed. The valid value is: • Handling overpayment through match type	Not applicable

   - **Preference Settings** - Used to define the settings for the overpayment handling through a match type preference.

Copyright © 2024, Oracle and/or its affiliates.

<table>
<tr><th>Field Name</th><th>Field Description</th><th>Mandatory (Yes or No)</th></tr>
<tr><td>Attribute</td><td>Used to indicate the attribute which you want to set in the overpayment handling through a match type preference. The valid values are:
• Match Type for On Account Payment (C) - At present, there is no functionality built using this attribute. It would be used in the future release of ORMB.
• Match Type for Default Payment Distribution (D) - Used to indicate the match types for which the system should behave in the following manner:
<table>
<tr><th>PUPL Batch Run</th><th>Then...</th></tr>
<tr><td>First</td><td>The system does the following:
• Processes the first payment record of each payee account and changes the status of the corresponding payment tender staging records to Pending.
• Does not process the subsequent payment records of each payee account and changes the payment tender staging record status of the subsequent payment records to Ignored.
Note: Once the three-step PUPL batch cycle is completed, the status of the payment record is changed to Processed or Error in Staging depending on whether it is successfully processed or not. In addition, the status of the corresponding payment tender staging records is changed from Pending to Complete.</td></tr>
</table></td><td>Yes (Conditional)
Note: This field is required while defining the settings for an overpayment handling through a match type preference.</td></tr>
<tr><td></td><td>
<table>
<tr><th>PUPL Batch Run</th><th>Then...</th></tr>
<tr><td>Second</td><td>The system does the following:
• Processes the second payment record of each payee account and changes the status of the corresponding payment tender staging records from Ignored to Pending.
• Does not process the subsequent payment records of each payee account and the payment tender</td></tr>
</table></td><td></td></tr>
</table>

Copyright © 2024, Oracle and/or its affiliates.

<table>
<tr><th>Field Name</th><th>Field Description</th><th>Mandatory (Yes or No)</th></tr>
<tr><td></td><td>
<table>
<tr><th>PUPL Batch Run</th><th>Then...</th></tr>
<tr><td></td><td>staging record status of the subsequent payment records remains as Ignored.</td></tr>
<tr><td></td><td>Note: Once the three-step PUPL batch cycle is completed, the status of the payment record is changed to Processed or Error in Staging depending on whether it is successfully processed or not. In addition, the status of the corresponding payment tender staging records is changed from Pending to Complete.</td></tr>
</table>
Note: In each three-step PUPL batch cycle, the system considers and processes one payment record of a payee account at a time and therefore you need to repeat the three-step PUPL batch cycle until all payment records of the payee account are processed.
</td><td></td></tr>
<tr><td></td><td>
• Match Type for Overpayment Distribution (O) - Used to indicate the match types for which the system should behave in the following manner:
<table>
<tr><th>PUPL Batch Run</th><th>Then...</th></tr>
<tr><td>First</td><td>The system does the following:
• Processes the first payment record of each payee account and match value combination and changes the status of the corresponding payment tender staging records to Pending.
• Does not process the subsequent payment records of each payee account and match value combination and changes the payment tender staging record status of the subsequent payment records to Ignored.</td></tr>
</table>
</td><td></td></tr>
</table>

Copyright © 2024, Oracle and/or its affiliates.

<table>
<tr><th>Field Name</th><th>Field Description</th><th>Mandatory (Yes or No)</th></tr>
<tr><td></td><td>
<table>
<tr><th>PUPL Batch Run</th><th>Then...</th></tr>
<tr><td></td><td>Note: Once the three-step PUPL batch cycle is completed, the status of the payment record is changed to Processed or Error in Staging depending on whether it is successfully processed or not. In addition, the status of the corresponding payment tender staging records is changed from Pending to Complete.</td></tr>
</table>
</td><td></td></tr>
<tr><td></td><td>
<table>
<tr><th>PUPL Batch Run</th><th>Then...</th></tr>
<tr><td>Second</td><td>The system does the following:
• Processes the second payment record of each payee account and match value combination and changes the status of the corresponding payment tender staging records from Ignored to Pending.
• Does not process the subsequent payment records of each payee account and match value combination and the payment tender staging record status of the subsequent payment records remains as Ignored.</td></tr>
<tr><td></td><td>Note: Once the three-step PUPL batch cycle is completed, the status of the payment record is changed to Processed or Error in Staging depending on whether it is successfully processed or not. In addition, the status of the corresponding payment tender staging records is changed from Pending to Complete.</td></tr>
</table>
Note: In each three-step PUPL batch cycle, the system considers and processes one payment record of each payee account and match value combination at a time and therefore you need to repeat the three-step PUPL batch cycle until all payment records of each payee account and match value combination are processed.
</td><td></td></tr>
</table>

Copyright © 2024, Oracle and/or its affiliates.

<table>
<tr><th>Field Name</th><th>Field Description</th><th>Mandatory (Yes or No)</th></tr>
<tr><td></td><td>

- **Match Type for Segment Level Overpayment Distribution (R)** - Used to indicate the match types for which the system should behave in the following manner:

<table>
<tr><th>PUPL Batch Run</th><th>Then...</th></tr>
<tr><td rowspan="2">First</td><td>The system does the following:
• Processes all payment records of the same or different payee account with the same or different match value and changes the status of the corresponding payment tender staging records to Pending.</td></tr>
<tr><td>Note: Once the three-step PUPL batch cycle is completed, the status of the payment record is changed to Processed or Error in Staging depending on whether it is successfully processed or not. In addition, the status of the corresponding payment tender staging records is changed from Pending to Complete.</td></tr>
</table>

</td><td></td></tr>
</table>

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: You can specify maximum of 20 match types separated by a comma (,) symbol for the above attributes in the field mapping. Now, while executing the step-three PUPL batch, the system considers the field mapping which is specified in the **Field Mapping for Match Type** option type of the **C1-PAYFLUPLD** feature configuration whenever there are multiple payment records of the payee account in the CSV file. In the following scenarios, the system behaves in the manner as if the match type is set to the **Match Type Default Payment Distribution (D)** attribute: • The value for the **Field Mapping for Match Type** option type is not defined in the **C1-PAYFLUPLD** feature configuration • The value specified for the Field Mapping for Match Type option type in the **C1-PAYFLUPLD** feature configuration is invalid • The match type received in the payment record is not specified against any of the attribute in the referred field mapping To implement this change, a new status named **Ignored** is added in the **PAY_TND_STG_ST_FLG** lookup field.	
Value	Used to specify the attribute value.	Yes (Conditional) **Note:** This field is required while defining the settings for an overpayment handling through a match type preference.
Entity Type	Used to indicate the type of entity for which the attribute is applicable. The valid values are: • Account • Adjustment • Billable Charge • Membership • Payment • Policy **Note:** At present, this data is stored for informational purposes only and is not considered in the business logic. If required, you can add more entity types to the **MAP_ENTITY_FLG** lookup field.	No

Tip: Alternatively, you can edit the details of an overpayment handling through a match type preference by clicking the **Edit** button in the **Field Mapping** zone.

Copyright © 2024, Oracle and/or its affiliates.

3. Modify the required details in the **Main** section.
4. Define, edit, or remove the attribute from the overpayment handling through a match type preference, if required.

> **Note:** You must set at least one attribute in an overpayment handling through a match type preference.

5. Click **Save**.
   The changes made to the overpayment handling through a match type preference are saved.

Deleting an Overpayment Handling through a Match Type Preference

Procedure

To delete an overpayment handling through a match type preference:

1. Search for an overpayment handling through a match type preference in the **Field Mapping** screen.
2. In the **Search Results** section, click the **Delete** (🗑) icon in the **Delete** column *c*orresponding to the overpayment handling through a match type preference that you want to delete.
   A message appears confirming whether you want to delete the overpayment handling through a match type preference.

> **Tip:** Alternatively, you can delete an overpayment handling through a match type preference by clicking the **Delete** button in the **Field Mapping** zone.

3. Click **OK**.
   The overpayment handling through a match type preference is deleted.

Copying an Overpayment Handling through a Match Type Preference

Instead of creating an overpayment handling through a match type preference from scratch, you can create a new overpayment handling through a match type preference using an existing overpayment handling through a match type preference. This is possible through copying an overpayment handling through a match type preference. On copying an overpayment handling through a match type preference, the details including the attributes are copied to the new overpayment handling through a match type preference. You can then edit the details, if required.

Prerequisites

To copy an overpayment handling through a match type preference, you should have:

- Overpayment handling through a match type preference (whose copy you want to create) defined in the application
- Value defined for the **Field Mapping for Match Type** option type in the **C1-PAYFLUPLD** feature configuration.
- Values defined in the **FIELD_CAT_FLG**, **FIELD_STATUS_FLG**, and **FIELD_NAME_FLG** lookup field.
- Values defined in the **PAY_TND_STG_ST_FLG** lookup field.

Procedure

To copy an overpayment handling through a match type preference:

1. Search for an overpayment handling through a match type preference in the **Field Mapping** screen.
2. In the **Search Results** section, click the **Duplicate** (⧉) icon in the **Duplicate** column *c*orresponding to the overpayment handling through a match type preference whose copy yo want to create.
3. Click **OK**.

   The **Field Mapping** screen appears. It contains the following sections:

   - **Main** - Used to specify basic details of the overpayment handling through a match type preference. It contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Preference	Used to specify the overpayment handling through a match type preference.	Yes

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
Description	Used to specify the description for the overpayment handling through a match type preference.	Yes
Detailed Description	Used to specify additional information about the overpayment handling through a match type preference.	No
Status	Used to indicate the status of the overpayment handling through a match type preference. The valid values are: • Active • Inactive	Yes
Preference Category	Indicates the feature for which the preference is designed. The valid values is: • Handling overpayment through Match Type	Not applicable

- **Preference Settings** - Used to set the attributes in the overpayment handling through a match type preference.

Field Name	Field Description	Mandatory (Yes or No)
Attribute	Used to indicate the attribute which you want to set in the overpayment handling through a match type preference. The valid values are: • **Match Type for On Account Payment (C)** - At present, there is no functionality built using this attribute. It would be used in the future release of ORMB. • **Match Type for Default Payment Distribution (D)** - Used to indicate the match types for which the system should behave in the following manner: **PUPL Batch Run** | **Then...** First | The system does the following: • Processes the first payment record of each payee account and changes the status of the corresponding payment tender staging records to **Pending**. • Does not process the subsequent payment records of each payee account and changes the payment tender staging record status of the subsequent payment records to **Ignored**.	Yes (Conditional) **Note:** This field is required while defining the settings for an overpayment handling through a match type preference.

Copyright © 2024, Oracle and/or its affiliates.

<table>
<tr><th>Field Name</th><th>Field Description</th><th>Mandatory (Yes or No)</th></tr>
<tr><td></td><td>
<table>
<tr><th>PUPL Batch Run</th><th>Then...</th></tr>
<tr><td></td><td>Note: Once the three-step PUPL batch cycle is completed, the status of the payment record is changed to Processed or Error in Staging depending on whether it is successfully processed or not. In addition, the status of the corresponding payment tender staging records is changed from Pending to Complete.</td></tr>
</table>
</td><td></td></tr>
<tr><td></td><td>
<table>
<tr><th>PUPL Batch Run</th><th>Then...</th></tr>
<tr><td>Second</td><td>The system does the following:
• Processes the second payment record of each payee account and changes the status of the corresponding payment tender staging records from Ignored to Pending.
• Does not process the subsequent payment records of each payee account and the payment tender staging record status of the subsequent payment records remains as Ignored.</td></tr>
<tr><td></td><td>Note: Once the three-step PUPL batch cycle is completed, the status of the payment record is changed to Processed or Error in Staging depending on whether it is successfully processed or not. In addition, the status of the corresponding payment tender staging records is changed from Pending to Complete.</td></tr>
</table>
Note: In each three-step PUPL batch cycle, the system considers and processes one payment record of a payee account at a time and therefore you need to repeat the three-step PUPL batch cycle until all payment records of the payee account are processed.
</td><td></td></tr>
<tr><td></td><td>• Match Type for Overpayment Distribution (O) - Used to indicate the match types for which the system should behave in the following manner:</td><td></td></tr>
</table>

Copyright © 2024, Oracle and/or its affiliates.

<table>
<tr><th>Field Name</th><th>Field Description</th><th>Mandatory (Yes or No)</th></tr>
<tr><td></td><td>
<table>
<tr><th>PUPL Batch Run</th><th>Then...</th></tr>
<tr><td>First</td><td>The system does the following:
• Processes the first payment record of each payee account and match value combination and changes the status of the corresponding payment tender staging records to Pending.
• Does not process the subsequent payment records of each payee account and match value combination and changes the payment tender staging record status of the subsequent payment records to Ignored.</td></tr>
<tr><td></td><td>Note: Once the three-step PUPL batch cycle is completed, the status of the payment record is changed to Processed or Error in Staging depending on whether it is successfully processed or not. In addition, the status of the corresponding payment tender staging records is changed from Pending to Complete.</td></tr>
</table>
</td><td></td></tr>
<tr><td></td><td>
<table>
<tr><th>PUPL Batch Run</th><th>Then...</th></tr>
<tr><td>Second</td><td>The system does the following:
• Processes the second payment record of each payee account and match value combination and changes the status of the corresponding payment tender staging records from Ignored to Pending.
• Does not process the subsequent payment records of each payee account and match value combination and the payment tender staging record status of the subsequent payment records remains as Ignored.</td></tr>
</table>
</td><td></td></tr>
</table>

Copyright © 2024, Oracle and/or its affiliates.

<table>
<tr><th>Field Name</th><th>Field Description</th><th>Mandatory (Yes or No)</th></tr>
<tr><td></td><td>
<table>
<tr><th>PUPL Batch Run</th><th>Then...</th></tr>
<tr><td></td><td>Note: Once the three-step PUPL batch cycle is completed, the status of the payment record is changed to Processed or Error in Staging depending on whether it is successfully processed or not. In addition, the status of the corresponding payment tender staging records is changed from Pending to Complete.</td></tr>
</table>
Note: In each three-step PUPL batch cycle, the system considers and processes one payment record of each payee account and match value combination at a time and therefore you need to repeat the three-step PUPL batch cycle until all payment records of each payee account and match value combination are processed.
</td><td></td></tr>
<tr><td></td><td>
• Match Type for Segment Level Overpayment Distribution (R) - Used to indicate the match types for which the system should behave in the following manner:
<table>
<tr><th>PUPL Batch Run</th><th>Then...</th></tr>
<tr><td>First</td><td>The system does the following:
• Processes all payment records of the same or different payee account with the same or different match value and changes the status of the corresponding payment tender staging records to Pending.
Note: Once the three-step PUPL batch cycle is completed, the status of the payment record is changed to Processed or Error in Staging depending on whether it is successfully processed or not. In addition, the status of the corresponding payment tender staging records is changed from Pending to Complete.</td></tr>
</table>
</td><td></td></tr>
</table>

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: You can specify maximum of 20 match types separated by a comma (,) symbol for the above attributes in the field mapping. Now, while executing the step-three PUPL batch, the system considers the field mapping which is specified in the **Field Mapping for Match Type** option type of the **C1-PAYFLUPLD** feature configuration whenever there are multiple payment records of the payee account in the CSV file. In the following scenarios, the system behaves in the manner as if the match type is set to the **Match Type Default Payment Distribution (D)** attribute: • The value for the **Field Mapping for Match Type** option type is not defined in the **C1-PAYFLUPLD** feature configuration • The value specified for the Field Mapping for Match Type option type in the **C1-PAYFLUPLD** feature configuration is invalid • The match type received in the payment record is not specified against any of the attribute in the referred field mapping To implement this change, a new status named **Ignored** is added in the **PAY_TND_STG_ST_FLG** lookup field.	
Value	Used to specify the attribute value.	Yes (Conditional) **Note:** This field is required while defining the settings for an overpayment handling through a match type preference.
Entity Type	Used to indicate the type of entity for which the attribute is applicable. The valid values are: • Account • Adjustment • Billable Charge • Membership • Payment • Policy **Note:** At present, this data is stored for informational purposes only and is not considered in the business logic. If required, you can add more entity types to the **MAP_ENTITY_FLG** lookup field.	No

Note: While defining an overpayment handling through a match type preference, you must set at least one attribute in the overpayment handling through a match type preference.

Copyright © 2024, Oracle and/or its affiliates.

> **Tip:** Alternatively, you can copy an overpayment handling through a match type preference by clicking the **Duplicate** button in the **Field Mapping** zone.

4. Enter the required details in the **Main** section.
5. Define, edit, or remove the attribute from the overpayment handling through a match type preference, if required.
6. Click **Save**.
   The new overpayment handling through a match type preference is created.

Viewing the Overpayment Handling through a Match Type Preference Details

Procedure

To view the details of an overpayment handling through a match type preference:

1. Search for an overpayment handling through a match type preference in the **Field Mapping** screen.
2. In the **Search Results** section, click the **Broadcast** (📡) icon corresponding to the overpayment handling through a match type preference *w*hose details you want to view.
   The **Field Mapping** zone appears.
3. View the details of the overpayment handling through a match type preference in the **Field Mapping** zone.

Maintaining Payment Events

A payment event is used to record when moneys are remitted and how the moneys are allocated amongst accounts. The topics in this section describe how to maintain payment events.

> **Note: The system creates most payment events behind-the-scenes.** Most payment events are created by the system when it uploads payments and when it creates automatic payments. You should only have to access the payment event transaction if you need to correct a payment event or add a payment event real-time. For information about how the system creates payment events, refer to The Big Picture of Payments.

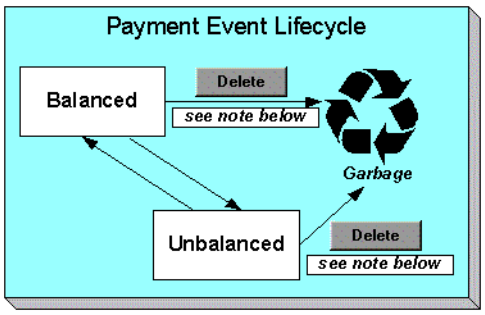
Payment Lifecycles

The topics in this section describe the lifecycle of the various payment objects.

Payment Event Lifecycle

The following diagram shows the possible lifecycle of a payment event.

> **CAUTION:** This diagram only makes sense in the context of the page used to maintain payment events. Refer to Payment Event - Main Information for the details.



The system, by default, distributes the sum of a payment event's tenders to the account that remits the tenders. After distribution the sum of the tenders equals the sum of the payments (remember, the term *payment* is used to refer to an

Copyright © 2024, Oracle and/or its affiliates.

allocation of some/all of a payment event's tenders to an account's debt) when the event is first created. We refer to such an event as being `Balanced`.

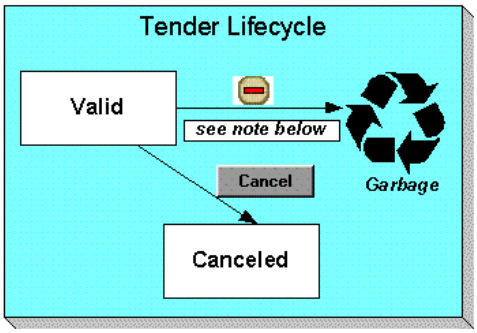
It is possible via any of the methods described in Unbalanced Payment Events to make a balanced payment event `Unbalanced`.

Click **Delete** to physically remove a balanced or unbalanced payment event from the database. You may not delete a payment event if: a) there are frozen or canceled payments linked to the event, or b) if there are canceled tenders linked to the event, or c) if a tender linked to the event is part of a balanced tender control. When the payment event is deleted, the system also deletes its tenders, payments, and payment segments.

Tender Lifecycle

The following diagram shows the possible lifecycle of a payment event.

CAUTION: This diagram only makes sense in the context of the page used to maintain tenders. Refer to Payment Event - Tenders for the details.



A tender is initially saved in the `Valid` state. It is possible via any of the methods described in Unbalanced Payment Events to make a balanced payment event `Unbalanced`.

If a tender is invalid, click **Cancel** to cancel the tender AND ALL PAYMENTS LINKED TO THE EVENT.

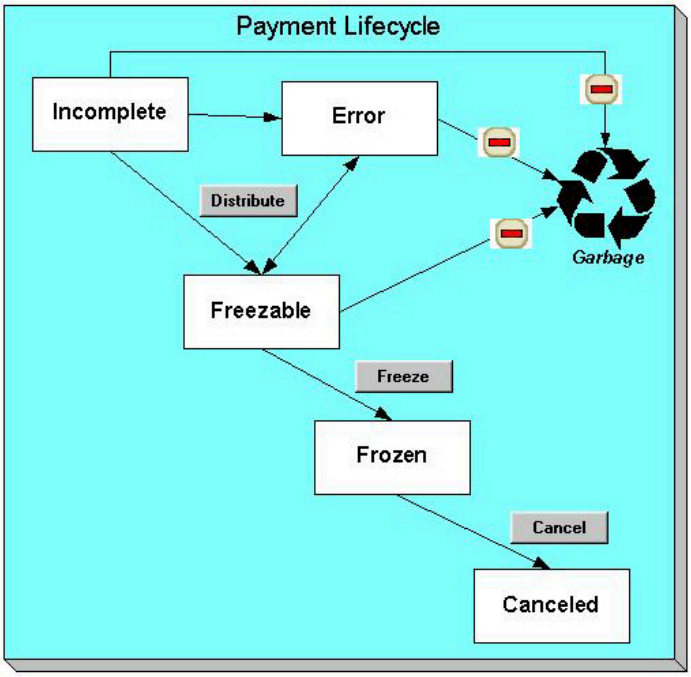
Click the delete button to physically remove a tender from the database. You may not delete a tender if it is part of a balanced tender control.

Payment Lifecycle

The following diagram shows the possible lifecycle of a payment.

CAUTION: This diagram only makes sense in the context of the page used to maintain a payment. Refer to Payment - Main for the details.

Copyright © 2024, Oracle and/or its affiliates.



Payments are initially created in the `Incomplete` state. Payments in this state don't have payment segments or financial transactions; they are simply a stub awaiting distribution.

Click **Distribute** to distribute a payment amongst an account's contracts.

- If the system cannot distribute the payment (for whatever reason), the payment is moved to the `Error` state. You may delete such a payment.
- If the system successfully distributes a payment, the payment becomes `Freezable`.

Click the - button to physically remove an `Incomplete`, `Error` or `Freezable` payment from the database.

Click **Freeze** to freeze a payment and its financial transaction. Freezing the payment causes the following to occur:

- The system executes any payment freeze algorithms linked to the account's customer class and to the contract's contract type.
- The payment's state becomes `Frozen` and the payment may now appear on a customer's bill.

You may not change a payment once it is frozen. However, you may reverse the payment's financial effect by clicking **Cancel**. Clicking this button will cause the following to occur:

- A new financial transaction is generated and linked to the payment. This financial transaction reverses the financial effects of the original payment.
- The system executes any payment cancellation algorithms linked to the account's customer class.
- If the payment has a related adjustment, this adjustment is also cancelled. Refer to Loan Overpayments for an example of payments with adjustments.
- The payment becomes `Canceled`.

Payment Event - Add Dialog

The Payment Event transaction features an unusual dialog that simplifies the addition of new payment events. This page appears if you open the **Payments, Payment Event** page in add mode from either the Account context menu or from the main menu (it also appears if you click the clear button when on the Payment Event page).

Note: If you have opted to always use the payment event distribution rules method as your default method, the Payment Event QuickAdd (Single Payment Event) page appears instead.

Copyright © 2024, Oracle and/or its affiliates.

Description of Page

The **Payor Account ID** is the account that remitted the payment. We assume this account is both the tendering account and the account whose debt is being relieved by the payment. If this assumption is not correct, choose a **Distribute Action** of `Do Not Distribute` and then change the tendering or paying account when the Payment Event page appears (after you click **OK**).

> **Note:** If you have navigated to this page from account context menu, the Payor Account ID and Payment Amount are defaulted to this account.

Select the **currency** when you want to apply the payment in the currency which is different from the account's invoice currency. You must always apply payments in the currency in which the bill is generated. This field is disabled when either of the following conditions is true:

- The **Allow Multi-Currency Accounts** option type of the **C1_MLTCURACC** feature configuration is set to **N**.
- The account's customer class is disabled for open item accounting. In other words, the **Open Item Accounting** check box is not selected for the account's customer class.

The **Payment Amount** is the amount of the customer's debt to be relieved by the payment. Note, this amount is defaulted using an algorithm plugged in on the Installation Record. Please refer to APAM-DFLT for more information about how the algorithm that is supplied with the base package calculates this amount.

The payment tenders grid allows you to enter multiple tender types and amounts. Click + to add a new tender. For each tender specify the following fields:

- The **Amount Tendered** is the amount of moneys remitted for the tender type.
- **Tender Type** describes what was remitted (e.g., cash, check, ...). **Tender Type** defaults from the Quick Add Tender Type that is defined on the installation record.
- If a check was tendered, use **Check Number** to specify the identity of the check.

> **Note: Cash back causes an additional tender to be created.** If cash should be returned to the customer (because the customer overpaid and the tender type's cash back allowed switch is true and the tender type is not "like cash"), a negative tender for the cash back amount is created for the payment event. Refer to Cash Back for a description of how the system can recommend Cash Backamount if the customer tendered more than they are paying.

Match Type and **Match Value** are used if either of the following conditions is true:

- This **Payor Account** belongs to an open item customer class. In this situation, specify a **Match Type** to define how the payment should be matched to the customer's open-items and use **Match Value** to define the open-items covered by the payment. For example, if this payment is in respect of a bill, specify a match type of "bill id" and a match value of the bill id being paid.

> **Note: Shortcut.** If you enter a **Match Type** of "bill id" and leave the **Match Value** blank, the system assumes the customer wants to pay the latest bill.

- The customer wants to restrict the distribution of the payment to a specific contract. In such case, select the **Contract ID** option from the **Match Type** list and enter the contract ID in the **Match Value** field.

The **Payment Date** defaults to the current date.

If the **Payor Account ID**'s customer class is designated as non-CIS (i.e., the person making the payment isn't a customer), the following information appears in the above window:

- **Non CIS Name** is the name of the person remitting the payment.
- **Reference Number** is the reference number of the item being paid (e.g., the property tax reference number).
- **Non CIS Comments** are used to describe anything unusual about the non-CIS payment.

Use **Distribute Action** to describe what you'd like to have happen when you click the **OK** button:

- Choose `Distribute and Freeze if OK` if this is a simple payment that should require no manual intervention. By "simple payment" we mean:

Copyright © 2024, Oracle and/or its affiliates.

  - The account is both the tendering account and the account whose debt is being relieved by the payment
  - The payment date is the current date
  - The payment should be distributed amongst the account's contracts using standard distribution logic
- If this option is selected, the system distributes the **Payment Amount** amongst the account's contracts. If the distribution is successful, the system automatically freezes the payment. If the distribution is not successful, the payment will be in the `Error` or `Incomplete` state. When the Payment Event page appears, you can view the error and then correct it. After the cause of the error is corrected, you must distribute and freeze the payment manually (this can be done on several pages including Payment Event - Main and Payment - Main).
- Choose `Manual Distribution` if you need to manually distribute the payment amongst the account's contracts. If this option is selected, the system creates a payment event, a tender and a payment and then transfers you to the Payment - Manual Distribution page where you can define the amount to be allocated to each of the account's contracts. After you've distributed the payment, don't forget to freeze it.
- Choose `Do Not Distribute` if you want to process the payment event manually (e.g., if you need to define multiple accounts whose debt is relieved by the payment). If this option is selected, the Payment Event - Main page opens with the information you entered defaulted accordingly. You can make any changes you want and then distribute and freeze the payment. Refer to How To Add A New Payment Event for more information.

Note: Cash only warning. If the account has exceeded your cash-only threshold and the **Tender Type** isn't marked as "like cash", a warning appears advising of such. Note, a customer's cash-only points are maintained on Account - Credit Rating.

Payment Event - Main Information

The Main page contains core payment event information. Open this page using **Payments, Payment Event**.

Fastpath: The **Description of Page** section below describes the fields on this page. Refer to How To for a description of how to perform common payment event maintenance functions.

Description of Page

Note: Cash-only customers. When you attempt to add a new payment event, the system warns you if the account remitting the tender is a cash-only customer and the tender's tender type is not cash. An account is considered cash-only if their current cash-only points exceed the cash-only tolerance maintained on the CIS Installation record. A customer's cash-only points are maintained on Account - Credit Rating.

Pay Event Info contains a concatenation of the payment date, amount, and the name of the main customer on the account that remits the tender. If multiple tenders exist, the customer's name is not displayed. If the payment event is associated with a single distribution detail, the rule name and the description of the rule value are displayed as well. If multiple distribution details exist, `Multiple Distribution Details Exist` is displayed instead. **Pay Event Info** is only displayed after the payment event has been added to the database.

Payment Event ID is the system-assigned unique identifier of the payment event.

The area under **Pay Event Info** provides warnings about the payment event. Possible warnings are `Unfrozen Payments` and `The Payment Event is Unbalanced`. A warning is also issued to the cashier to remind him/her to turn in funds - see Turn Ins.

If multiple distribution details are linked to the payment event, the distribution rule, value, and amount for each distribution detail displays.

Payment Date is the business date associated with the payment event.

CAUTION: If you change the payment date and this event's tender(s) are automatic payments, the extract date (i.e., the date the automatic payment is sent to the financial institution) will be changed to equal the payment date. Refer to Automatic Payments **for more information**.

Payment(s) contains the total number and value of payments linked to the event.

Copyright © 2024, Oracle and/or its affiliates.

Tender(s) contains the total number and value of tenders linked to the event.

Amount Tendered contains the amount that was tendered by the customer.

The system displays a **Cash Back Amount** if the customer tenders more than they are paying (refer to Cash Back for details).

If the payment event becomes `unbalanced`, a **Recalculate Cash Back** button appears. If this button is clicked, the system creates, deletes or recalculates the cash back tender. Refer to Cash Back for details.

While most payment events contain a single payment, the system allows many payments to exist under a payment event (when the payment event's tender(s) are distributed amongst multiple accounts). If a payment event has a large number of payments, you can use the **Account Filter** to limit the payments that appear in the **Payments** grid. The following options are available:

- `Account.` Use this option if you only want to see the payment linked to an **Account ID**.
- `All.` Use this option to view all payments linked to the payment event.
- `Person Name.` Use this option to restrict payments linked to accounts whose main customer has a primary name that matches **Person Name**.

The **Payments** grid contains the payment(s) linked to the payment event. The following points describe the attributes in this scroll; refer to How To for instructions describing how to perform common maintenance activities.

- **Account ID** references the payment's account. The name of the account's main customer is displayed adjacent.
- **Payment Amount** is the amount of the account's debt relieved by the payment. The adjacent context menu allows you to drill down to the details of the payment (this is where you can see the payment's payment segments and where you can override the distribution of the payment).
- **Payment Status** is the payment's status. If the payment's status is `Error`, the error message is displayed adjacent. Refer to Payment Lifecycle for the potential values and how to handle a payment when it exists in a given state.
- **Match Type** and **Match Value** should only be used if either of the following conditions is true:
  - This **Account ID** belongs to an open item customer class. In this situation, specify a **Match Type** to define how the payment should be matched to the customer's open-items and use **Match Value** to define the open-items covered by the payment. For example, if this payment is in respect of a bill, specify a match type of "bill id" and a match value of the bill id being paid.
  - The customer wants to restrict the distribution of the payment to a specific contract. In such case, select the **Contract ID** option from the **Match Type** list and enter the contract ID in the **Match Value** field.
- The remaining columns are only used if the payment is linked to an **Account ID** that belongs to a customer class that is used for non-CIS payments. Refer to Setting Up Customer Classes for more information. If such an account exists, the following fields must be defined.
  - **Non CIS Name** is the name of the person remitting the payment.
  - **Reference Number** is the reference number of the item being paid (e.g., the property tax reference number).
  - **Non CIS Comments** are used to describe anything unusual about the non-CIS payment.
- Note, you can also define a comment for a non-CIS payment. To define a comment, use the context menu adjacent to **Payment Amount** to drill to Payment - Main.
- **Payment ID** is the unique, system-assigned identifier of the payment. This value only appears after the payment has been added to the database.

Refer to Payment Actions and Payment Event Actions for information about the action buttons on this page. Refer to How To for a description of typical business processes that use these buttons.

Payment Event - Tenders

The Tenders page contains a scroll showing one row for every tender associated with the payment event. Open this page using **Payments, Payment Event, Tenders**.

Fastpath: The **Description of Page** section below describes the fields on this page. Refer to How To for instructions describing how to perform common maintenance functions.

Copyright © 2024, Oracle and/or its affiliates.

Description of Page

Pay Event Info contains a concatenation of the payment date, amount, and the name of the main customer on the account that remits the tender. If multiple tenders exist, the customer's name is not displayed. If the payment event is associated with a single distribution detail, the rule name and the description of the rule value are displayed as well. If multiple distribution details exist, `Multiple Distribution Details Exist` is displayed instead. **Pay Event Info** is only displayed after the payment event has been added to the database.

Payment Event ID is the system-assigned unique identifier of the payment event.

Payment Date is the business date associated with the payment event.

Payment(s) contains the total number and value of payments linked to the event.

Tender(s) contains the total number and value of tenders linked to the event.

Amount Tendered contains the amount that was tendered by the customer.

The system displays a **Cash Back Amount** if the customer tenders more than they are paying (refer to Cash Back for details).

If the payment event becomes `unbalanced`, a **Recalculate Cash Back** button appears. If this button is clicked, the system creates, deletes or recalculates the cash back tender. Refer to Cash Back for details.

The **Tenders** scroll controls the display of the tenders linked to the payment event. The following simply describes the fields in this scroll; refer to How To for instructions describing how to perform common tender maintenance activities.

- **Payor Account ID** references the tendering account. The name of the account's main customer is displayed adjacent. If the account has exceeded your cash-only threshold, the warning `Cash only` appears.

CAUTION: If you change the Payor Account ID and the tender has an associated automatic payment request, the automatic payment request will be removed. A new automatic payment request will be created for the new Payor Account ID. Refer to Automatic Payments **for more information**.

- **Tender Amount and Currency** field displays the tender amount and currency.
- **Original Tender Amount and Currency** field displays the tender amount and currency before currency conversion. This field appears when the currency conversion is done during payment and tender creation.
- **Exchange Rate** field displays the exchange rate value used during the currency conversion. This field appears when the currency conversion is done during payment and tender creation.
- **Exchange Rate ID** field displays the ID that uniquely identifies the exchange rate used during the currency conversion. This field appears when the currency conversion is done during payment and tender creation.
- If a check was used, the **Check Number** contains the identity of the check.
- **Pay Tender ID** is the system-assigned unique identifier of the tender. This value appears after the tender has been added to the database.
- **Tender Type** describes what was remitted (e.g., cash, check). If the **Tender Type** is associated with an automatic payment, the page displays a section that includes information about how and when the automatic payment was interfaced to the payment source.

Note: Authorizing Tenders. The **Tender Type** dropdown list does not include tender types that require authorization (i.e. credit card payments).

Fastpath: For information about tender types requiring authorization, refer to Credit Card Payments.

- **Tender Status** is the tender's status. Refer to Tender Lifecycle for the potential values and how to handle a tender when it exists in a given state.
- If the **Tender Type** is associated with an automatic payment, the system attempts to default automatic payment information from the account's auto-pay option if the tender type is the same as the tender type on the account's auto-pay source and if the auto pay option is effective on the payment date. If the system is unable to default information, you must specify the source of the funds and the customer's account number / credit card number at the financial institution. You can override automatic payment information as long as the automatic payment has not been sent to the financial institution or cancelled. The **Auto Pay Section** contains the following fields:

Copyright © 2024, Oracle and/or its affiliates.

  - **Auto Pay Source Code** is the financial institution / credit card company that receives the automatic payment request.
  - **Scheduled Extract Date** is the date the automatic payment request was sent / is scheduled to be sent to the financial institution. This information is display-only.
  - **External Account ID** is the customer's account number at the financial institution.
  - **Expires On** is only needed if the Tender Type indicates that an expiration date is necessary (e.g., for a credit card payment).
  - **Name** is the customer's name in the financial institution's system.
  - **Bill ID** is the ID of the bill whose completion caused the creation of the automatic payment. This information only appears if the automatic payment was created as a result of a bill. This information is display-only.
- If the payment was uploaded, the following fields MAY contain information (if they were populated in the interface file):
  - **MICR ID** is the value of the magnetic ink character recognition (MICR) line on the payment.
  - **Customer ID** is the customer's account ID that appeared on the interfaced payment.
  - **Ext. Reference ID** is the unique identifier of the payment upload interface record.
  - **Name** is the customer's name on the interfaced payment record.
- In the **Tender Control** area, the **Tender Control ID** is the identity of the remittance processor batch or cash drawer bundle in which the tender was remitted. Refer to Tender Management and Workstation Cashiering for more information about tender controls.
- Displayed adjacent to **Tender Control** is the date, tender source, and status of the tender control.

Every new tender must reference an `open` tender control. The system will attempt to default an appropriate tender control as follows:

- If the user has at least one `open`, user-specific Tender Control, the system will default one, at random.
- If the user has no `open`, user-specific Tender Controls; the system will default an open, "all users" tender control if it can find one whose tender source indicates it's used for online cashiering. If multiple exist, one will be selected at random.
- Turn off **Included in Tender Ctl Balance** if the tender should not be included in the tender amount in the tender control's tender balance. You would turn this switch off if you canceled a tender because it was mistakenly applied to the wrong account (or the amount was wrong). This switch is only enabled if:
  - The tender control isn't `balanced`.
  - The tender is `canceled`.
- **Deposit Control** contains a concatenation of the tender control's deposit control's creation date, tender source type, and status. Refer to Tender Management and Workstation Cashiering for more information about deposit controls.
- The **Tender Action** area contains a button that you use to cancel a tender. Refer to Tender Actions for information about this button. Refer to How To Cancel A Tender for more information.

The **Characteristics** collection on the Tenders page contains information that describes miscellaneous information about the tender. The following fields display:

Characteristic Type Indicate the type of characteristic.

Sequence Controls the order in which characteristics of the same type are displayed.

Characteristic Value Indicate the value of the characteristic.

Payment Event Action Codes

The topics that follow describe each of the actions that appear on the payment event pages.

Payment Actions

The topics that follow describe payment-oriented actions. Refer to How To for a description of several business processes that use these buttons.

Copyright © 2024, Oracle and/or its affiliates.

Distribute (Payments)

Clicking the **Distribute** button distributes all `Incomplete`, `Error`, or `Freezable` payments linked to the event. Refer to Distribute (A Payment) for information of how a single payment is distributed.

This button is enabled if at least one payment is `Incomplete`, `Error`, or `Freezable`.

Redistribute (Payments)

Clicking the **Redistribute** button redistributes all `Incomplete`, `Error`, or `Freezable` payments linked to the event. Refer to Redistribute (A Payment) for information of how a single payment is redistributed.

This button is enabled if at least one payment is `Incomplete`, `Error`, or `Freezable`.

Print

This button is enabled under the following conditions:

- The document `cm_POSPrint.js` must be present on the web server and the variable defined in that document must be set to `true`. This is to indicate that point of sale (POS) printers are in use. Refer to the installation guide for more information about setting up receipt printers.
- The payment event must have at least one payment that is `frozen` AND no payments that are in `Incomplete`, `Error` or `Freezable`

Refer to How To Print Receipts And Endorsements for more information.

Freeze (Payments)

Clicking the **Freeze** button causes all `Freezable` payments linked to the event to become `frozen`. Refer to Freeze (A Payment) for information of how a single payment is frozen.

This button is enabled if at least one payment is `Freezable`.

Note: If problems are detected after freezing. A payment may not be changed after it is `frozen`. All subsequent changes must occur by canceling the frozen payment and creating a new one. You can cancel and redistribute a payment on the next tab. Refer to Canceling A Tender Versus Canceling a Payment for more information.

Payment Event Actions

The topics that follow describe payment event-oriented actions. Refer to How To for a description of typical business processes that use these buttons.

Transfer

If the payment event was created using distribution rule(s), any transfer of its payments should also be done using the distribution rules method. Hence, a different dialog opens when you click **Transfer** based on whether the payment event is associated with at least one distribution detail or not.

Transfer (No Distribution Details Exist)

The **Transfer** button is enabled if:

- There is a single non-cancelled tender linked to the event AND
- There is a single frozen payment linked to the event AND
- The account on the tender is the same as the account on the payment.

When clicked, the **Payment Event Transfer** dialog opens.

You must specify the following parameters in the **Payment Event Transfer** dialog to transfer a payment:

- **Account ID** is the account to which the payment event should be transferred.
- **Cancel Reason** defines why you are performing the transfer.
- **Match Type** and **Match Value** are defaulted based on their values for the payment event that you are transferring. If necessary, modify the values appropriately for the account to which you are transferring the payment event. Match

Copyright © 2024, Oracle and/or its affiliates.

Type and Match Value are used for an account that belongs to an open item customer class or to restrict the distribution of the payment to a specific contract. Refer to Payment Event - Main for more information.

- Turn on **Freeze Payment** if the transferred payment should be frozen automatically. You may want to leave this turned off if you want to examine the payment distribution before freezing the payment.
- **Non CIS Name**, **Reference Number**, and **Non CIS Comments** appear if the **Account ID** belongs to a customer class that is used for non-CIS payments. Refer to Setting Up Customer Classes for more information.

Clicking **OK** causes the following to take place:

- The account on the tender is changed to reflect the transfer to account.
- The original payment and its segments are cancelled.
- A new payment is added for the transfer to account. The payment characteristics associated with the original payment are copied to the new payment.
- The new payment is distributed.
- If so designated, the new payment is frozen.

Transfer (Distribution Details Exist)

The **Transfer** button is enabled if:

- There is a single payor AND
- All existing payees before the transfer are the same as the payor AND
- There is at least a single non-cancelled tender linked to the event

When clicked, the **Payment Event Transfer** dialog opens.

You must specify the following parameters in this dialog to transfer payment(s):

- **Distribution Rule** is the account to which the payment event should be transferred.
- **Rule Value** is the value associated with the payment and expected by the distribution rule.
- **Amount** is the payment amount.
- **Cancel Reason** defines why you are performing the transfer.

Clicking **OK** causes the following to take place:

- The payment(s) associated with the original account are canceled.
- The new set of distribution details are processes, creating new payment(s) for the transfer to account. The payment characteristics associated with the original payment are copied to the new payment.
- If all new payments are for the same payee account (and the payee is different than the current payor) the account on the tender is changed to reflect the transfer to account.
- Original distribution details (only those with a non-zero amount) are deleted and replaced by the new set of details on the payment event.

Note: Distribution rule entries can have a zero amount. Refer to Rule Value Can Capture Additional Information for more information about how to use distribution details to capture additional payment related information.

Note: Determine Tender Account. This dialog does not call the `Determine Tender Account` algorithm defined on distribution rule as this action does not rebuild the payment tender collection.

Fastpath: Refer to Rule Value Can Capture Additional Information for more information about how to use distribution details to capture additional payment related information.

Delete A Payment Event

The **Delete** button deletes a payment event and its tenders and payments.

This button is enabled if there are no `frozen` or `canceled` payments and no `canceled` tenders.

Copyright © 2024, Oracle and/or its affiliates.

Tender Actions

The topics that follow describe tender-oriented actions. Refer to How To for a description of several business processes that use these buttons.

Cancel A Tender

Clicking **Cancel** causes the following to take place:

- All payments associated with the tender's payment event are `canceled`.
- The tender becomes `canceled`.
- If the cancellation reason indicates the cancellation is due to non-sufficient funds, other actions may occur. Refer to NSF Cancellations for more information.

The **Cancel** button is enabled if the tender is not `canceled`

When clicked, the **Tender Cancel** window opens.

You must specify the following parameters to cancel a tender:

- Select a **Cancel Reason** to describe why the tender is being canceled.
- The system also needs to know which **Bank Account To Charge** the cancellation against. The system will default the **Bank Code** and **Bank Account** from the original bank information used when the tender was deposited. You can override them here.

When you click the **OK** button, the system cancels the tender and ALL payments linked to the event.

Note: When non-canceled tenders are still linked to the event. If there are multiple tenders linked to the payment event, you must either add new payment(s) that equal the sum of the event's non-canceled tenders or cancel the remaining tenders.

Note: Cancellation after cash back. If a customer tenders a check for $100, but only owes $80, the system will recommend returning $20 in cash to the customer (assuming the tender type for checks allows overpayment). If the tender type for checks is not "like cash", a second tender is created for -$20 (the first tender is for $100). If the check subsequently bounces, both tenders must be cancelled.

Payment Event QuickAdd

The Payment Event QuickAdd page is used to quickly add, distribute and freeze one or more payment events using distribution rules. Open this page using **Payments, Payment Event QuickAdd**.

Description of Page

Specify the **Tender Control ID** in which the tenders will be recorded. Every new tender must reference an `open` tender control. The system will attempt to default an appropriate tender control as follows:

- If the user has at least one `open`, user-specific Tender Control whose type is online cashiering or ad hoc, the system will default one, at random.
- If the user has no `open`, user-specific Tender Controls; the system will default an open, "all users" tender control if it can find one whose tender source whose type is online cashiering or ad hoc. If multiple exist, one will be selected at random.

Specify the **Payment Date**. The current date defaults.

Use **Number of Payment Events** to indicate whether you intend to add a `single` payment event or `multiple` payment events.

Choose the `multiple` payment events option if you intend to add "simple" payment events. By "simple" we mean:

- A single account is both the payor account and the account whose debt is being relieved by the payment.
- The payment tender is not an auto-pay tender.

Copyright © 2024, Oracle and/or its affiliates.

For all other cases choose the `single` payment event dialogue as it is designed to handle more complex payment event configurations such as when:

- The payor account is different than the payee account.
- Multiple payors cover payments for one or more payees.
- The payment tender is an auto-pay tender.

Once you have made your selection the page is adjusted to support the corresponding dialog.

Note: Cash-only customers. When you attempt to add a new payment event, the system warns you if the account remitting the tender is a cash-only customer. An account is considered cash-only if their current cash-only points exceed the cash-only tolerance maintained on the Installation record. A customer's cash-only points are maintained on Account - Credit Rating.

Multiple Payment Events Dialog

This section describes the default dialog setup when navigating from **Payments, Payment Event QuickAdd.**

Description of Page

Using the `multiple` payment events option, each row in the payment detail grid typically represents a unique distribution detail. However, more than one row in the grid can belong to a single payment event. All distribution details for identical tender account will be grouped under a single pay event.

The following columns appear in the **Payments** grid:

- Select a **Distribution Rule** by which the payment detail is to be processed. If you have set up a default distribution rule, it is defaulted in the first row.
- Specify the **Rule Value** associated with the payment and expected by the distribution rule.
- Use **Payment Amount** to define the amount of the payment.
- Use **Tender Type** to define the form or remittance (e.g., cash, check, etc.). Note that the **Tender Type** defaults from the installation record.

Note: Authorizing Tenders. The **Tender Type** dropdown list does not include tender types that require authorization (i.e. credit card payments).

Fastpath: For information about tender types requiring authorization, refer to Credit Card Payments.

Note: Automatic Payments. The multiple payment events dialog does not support automatic payments. Switch to the single payment event dialog to enter automatic payments.

- Use **Check Number** if a check is remitted.
- **MICR ID** is the value of the magnetic ink character recognition (MICR) line on the payment.
- You may use **Customer ID** to record additional person information.
- You may use **External Reference ID** to record external information associated with the payment tender.
- You may use **Name** to record additional payment tender information.
- **MICR ID** is the MICR ID associated with the tender.
- **Payor Account ID** is the tender account as determined by the `Determine Tender Account` algorithm defined on the distribution rule. This information is only populated after the distribution detail has been processed.

After specifying the various payment distribution details in the grid, click **Create**. The system attempts to create payment event(s) as follows:

- As mentioned earlier, all distribution details for identical tender accounts are grouped under a single pay event. The first step in identifying common tenders is to determine the tender account. To do that the system calls the `Determine Tender Account` algorithm defined on the distribution rule.
- A payment event is created for each unique tender account.

Copyright © 2024, Oracle and/or its affiliates.

- All rows having all attributes of the tender identical except for the amount are grouped together where each group represents a single tender.
- A single payment tender is created for each unique tender and linked to the payment event.
- The total payment amount for each distinct group of distribution details (for the payment event) having the same rule and value is summarized. For each distinct group, the system calls the `Create Payment` algorithm defined on the distribution rule providing it with the rule value and total payment amount. It is the responsibility of this algorithm to create the payments for this payment event.
- Add the distinct payment event distribution detail and its total amount beneath the payment event.
- If the tender control's tender source type indicates that this is a cashiering station and your implementation is configured for printing at a cashier station, the Print Dialog appears.

The **Payment Event ID** column appears in the grid after the distribution details have been processed. It shows the payment event ID created for the distribution detail record and a short description as to the status of the payment(s) created for the payment event. You can use this field to navigate to the payment event.

After you've added a group of payment events, you should press the clear button (or Alt-C), to ready the page for the next group of payment event details.

Single Payment Event Dialog

If you have opted to always use the payment event distribution rules method as your default method, this section describes the default dialog setup when you navigate to the **Payment Event** page in **Add** mode.

Adding a `single` payment event at a time, allows for more complex payment event information to be captured.

Enter one row in the **Tenders** grid for every unique tender associated with the payment event.

- **Payor Account ID** references the tendering account. The name of the account's main customer is displayed adjacent. If the account has exceeded your cash-only threshold, the warning `Cash only` appears.

CAUTION: If you change the Payor Account ID and the tender has an associated automatic payment request, the automatic payment request will be removed. A new automatic payment request will be created for the new Payor Account ID. Refer to Automatic Payments **for more information**.

- **Tender Amount** is the amount of the tender.
- Use **Tender Type** to define the form or remittance (e.g., cash, check, etc.). Note, the **Tender Type** defaults from the installation record.
- Use **Check Number** if a check is remitted.
- **MICR ID** is the value of the magnetic ink character recognition (MICR) line on the payment.
- You may use **Customer ID** to record additional person information.
- You may use **External Reference ID** to record external information associated with the payment tender.
- You may use **Name** to record additional payment tender information.
- If the **Tender Type** is associated with an automatic payment, the page displays a section that includes information about how and when the automatic payment was interfaced to the payment source.
- If the **Tender Type** is associated with an automatic payment, the system attempts to default automatic payment information from the account's auto-pay option if the tender type is the same as the tender type on the account's auto-pay source and if the auto pay option is effective on the payment date. If the system is unable to default information, you must specify the source of the funds and the customer's account number / credit card number at the financial institution.
    - **Auto Pay Source Code** is the financial institution / credit card company that receives the automatic payment request.
    - **External Account ID** is the customer's account number at the financial institution.
    - **Expires On** is only needed if the **Tender Type** indicates that an expiration date is necessary (e.g., for a credit card payment).
    - **Name** is the customer's name in the financial institution's system.

Copyright © 2024, Oracle and/or its affiliates.

The following columns appear in the **Payments** grid:

- Select a **Distribution Rule** by which the payment detail is to be processed. If you have set up a default distribution rule, it is defaulted in the first row.
- Specify the **Rule Value** associated with the payment and expected by the distribution rule.
- Use **Payment Amount** to define the amount of the payment.

After specifying the various payment distribution details in the grid, click **Create**.

Note: Go To Payment Event. If you wish to be transferred to the payment event page once processing is complete, remember to check the Go To Payment Event check box before you click **Create**. This field does not appear when this page is used as the default Payment Event - Add Dialog and you are automatically transferred to the payment event page once the payment event is created.

After distribution details have been processed, a short description of the payment event that has been created is displayed to the right of the **Go To Payment Event** field. The description provides information as to the status of the payment(s) created for the payment event. You can use this field to navigate to the payment event.

The system attempts to create the payment event as follows:

- Validate that the sum of the tenders equals the sum of the payments.
- Create the payment event with a separate payment tender for each row in the Tenders grid.
- Summarize the total payment amount for each distinct group of distribution details having the same rule and value. For each distinct group, call the `Create Payment` algorithm defined on the distribution rule providing it with the rule value and total payment amount. It is the responsibility of this algorithm to create the payments for the payment event.
- Add the distinct payment event distribution detail summary beneath the payment event.
- If the tender control's tender source type indicates that this is a cashiering station and your implementation is configured for printing at a cashier station, the Print Dialog appears.

Note: Determine Tender Account. Having an explicit collection of tenders eliminate the need to determine the tender account. Therefore this dialog does not call the `Determine Tender Account` algorithm.

Payment Quick Add

The Payment Quick Add page is used to quickly add, distribute and freeze payment events for up to fifteen accounts. Open this page using **Payments, Payment Quick Add**.

Description of Page

Specify the **Tender Control ID** in which the tenders will be recorded. Every new tender must reference an `open` tender control. The system will attempt to default an appropriate tender control as follows:

- If the user has at least one `open`, user-specific Tender Control whose type is online cashiering or ad hoc, the system will default one, at random.
- If the user has no `open`, user-specific Tender Controls; the system will default an open, "all users" tender control if it can find one whose tender source whose type is online cashiering or ad hoc. If multiple exist, one will be selected at random.

Specify the **Payment Date**. The current date defaults.

Enter one row in the grid for every payment event to be added. The following information is entered for each payment event:

- Use **Account ID** to define the customer who tendered the payment. It's important to note that this transaction assumes that the tendering account is the same as the account whose balance is relieved by the payment. Refer to How To Allocate The Tender Amount To Multiple Accounts if you need to distribute the payment to an account (or accounts) other than the tendering account.

Copyright © 2024, Oracle and/or its affiliates.

Note: Cash-only customers. You cannot use this transaction to add a payment for a "cash-only" account if the tender type is not "like cash". An account is considered cash-only if its cash-only points exceed the cash-only tolerance maintained on the Installation record. A customer's cash-only points are maintained on Account - Credit Rating. You must use the Payment Event transaction to add such a payment (note, if you use the Account context menu to transfer to the Payment Event page in add mode, you won't have to retype the Account ID when you add the payment event).

- Select the **currency** when you want to apply the payment in the currency which is different from the account's invoice currency. You must always apply payments in the currency in which the bill is generated. This field is disabled when either of the following conditions is true:
  - The **Allow Multi-Currency Accounts** option type of the **C1_MLTCURACC** feature configuration is set to **N**.
  - The account's customer class is disabled for open item accounting. In other words, the **Open Item Accounting** check box is not selected for the account's customer class.
- Use **Payment Amount** to define the amount of the payment.
- Use **Tender Type** to define the form or remittance (e.g., cash, check, etc.). Note, the **Tender Type** defaults from the installation record.

Note: Authorizing Tenders. The **Tender Type** dropdown list does not include tender types that require authorization (i.e. credit card payments).

Fastpath: For information about tender types requiring authorization, refer to Credit Card Payments.

- Use **Check Number** if a check is remitted.
- **Match Type** and **Match Value** are only used if either of the following conditions is true:
  - This **Account ID** belongs to an open item customer class. In this situation, specify a **Match Type** to define how the payment should be matched to the customer's open-items and use **Match Value** to define the open-items covered by the payment. For example, if this payment is in respect of a bill, specify a match type of "bill id" and a match value of the bill id being paid.

Note: Shortcut. If you enter a **Match Type** of "bill id" and leave the **Match Value** blank, we assume the customer wants to pay the latest bill.

- The customer wants to restrict the distribution of the payment to a specific contract. In such case, select the **Contract ID** option from the **Match Type** list and enter the contract ID in the **Match Value** field.

After specifying the various accounts and amounts in the grid, click the **Distribute and Freeze** button. When you click this button, the system attempts to create a payment event for each row in the grid. Four potential outcomes are possible for each row:

- If the data you entered for a payment event isn't complete (e.g., you don't specify a valid account or amount):
  - No payment event is created.
  - A **Message** describing the problem is displayed.
  - All fields on the row remain modifiable.
- You should correct each such line and then press the **Distribute and Freeze** button.
- If the data you entered is complete, but the system issues a warning (e.g., the account is considered "cash-only" and a tender type other than cash is remitted, or if the account already has a payment for this amount on this date):
  - No payment event is created.
  - A **Message** containing the warning is displayed.
  - All fields on the row remain modifiable.
- You must clear the line and add a payment using the Payment Event transaction (note, if you use the Account context menu to transfer to the Payment Event page in add mode, you won't have to retype the Account ID when you add the payment event).
- If the data you entered is complete, but the system is not successful in distributing the payment:
  - The system creates a payment event, a payment and a tender.

Copyright © 2024, Oracle and/or its affiliates.

  - A **Message** describing the problem is displayed.
  - The payment's **Status** is displayed
  - All fields on the row are disabled.
- You must press the adjacent go to button to drill to the payment event where the error can be corrected.
- If the data you entered is complete and the system is successful in distributing the payment:
  - The system creates a payment event, a payment, and a tender.
  - The system distributes the payment amongst the account's contracts and then freezes the payment.
  - The payment's **Status** is displayed.
  - All fields on the row are disabled.
  - If the tender control's tender source type indicates that this is a cashiering station and your implementation is configured for printing at a cashier station, the Print Dialog appears.
- You can press the adjacent go to button to view a payment event.

Note: Separate Commits. This page is unusual in that each payment event is committed to the database independently. For example, if you enter seven payment events and one is invalid, six payment events will be added to the database when you press **Distribute and Freeze**. When the page is redisplayed, the rows containing the valid payments are protected and an indication of their validity is displayed. The row containing the invalid payment remains unprotected. You can correct the erroneous payment and then press the **Distribute and Freeze** button again.

After you've added a group of payments, you should press the clear button (or Alt-C), to ready the page for the next group of payment events.

Maintaining Payments

Payments and payment segments are automatically created by the system when payment events are created. You should only need to access this transaction as follows:

- To view a payment's payment segments (i.e., to view how a payment was distributed amongst its account's contracts).
- To override the distribution of a payment amongst its account's contracts. Refer to Distributing A Payment Amongst An Account's Contracts for information describing how payments are typically distributed.

The topics in this section describe how to maintain a payment.

Payment - Main

This page contains basic information about a payment. Open this page using **Payments, Payment, Main**.

CAUTION: Most payments are maintained using the Payment Event page. The transaction described below is typically only used to view a payment's payment segments and to manually distribute a payment amongst an account's contracts.

Fastpath: The **Description of Page** section below describes the fields on this page. Refer to How To for instructions on how to perform common maintenance functions.

Description of Page

Payment Info contains a concatenation of the payment date, amount, status, and the name of the main customer on the account.

Payment ID is the system-assigned unique identifier of the payment.

Payment Event ID is the system-assigned unique identifier of the payment event. The adjacent info contains a concatenation of the payment date, amount, and the name of the main customer on the account that remits the event's tender. If multiple tenders exist, the customer's name is not displayed. A warning may also be displayed. The possible warnings are `Unfrozen Payments` and `The Payment Event is unbalanced`.

Copyright © 2024, Oracle and/or its affiliates.

Payment(s) contains the total number and value of payments linked to the payment event.

Tender(s) contains the total number and value of tenders linked to the payment event.

Account ID is the account whose debt is reduced by the payment. The name of the account's main customer is displayed adjacent. This field is gray after the payment is frozen.

Select the **currency** when you want to apply the payment in the currency which is different from the account's invoice currency. You must always apply payments in the currency in which the bill is generated. This field is disabled when either of the following conditions is true:

- The **Allow Multi-Currency Accounts** option type of the **C1_MLTCURACC** feature configuration is set to **N**.
- The account's customer class is disabled for open item accounting. In other words, the **Open Item Accounting** check box is not selected for the account's customer class.

Payment Amount is the amount of the payment. This field is gray after the payment is frozen. If the account has exceeded your cash-only threshold, a warning (in red) appears adjacent to **Payment Amount**. Note, a customer's cash-only points are maintained on Account - Credit Rating.

Payment Status is the payment's status. Refer to Payment Lifecycle for the potential values and how to handle a payment when it exists in a given state. If the payment is in `Error`, the error message is displayed adjacent. If the payment is `canceled`, the **Cancel Reason** is displayed adjacent.

Match Type and **Match Value** will only be used if either of the following conditions is true:

- This **Account ID** belongs to an open item customer class. In this situation, specify a **Match Type** to define how the payment should be matched to the customer's open-items and use **Match Value** to define the open-items covered by the payment. For example, if this payment is in respect of a bill, specify a match type of "bill id" and a match value of the bill id being paid.
- The customer wants to restrict the distribution of the payment to a specific contract. In such case, select the **Contract ID** option from the **Match Type** list and enter the contract ID in the **Match Value** field.

Match Type and **Match Value** are gray after the payment is frozen.

If the payment is linked to an **Account ID** that belongs to a customer class that is used for non-CIS payments, the following fields appear (note, these fields are gray after the payment is frozen):

- **Name** is the name of the person remitting the payment.
- **Reference Number** is the reference number of the item being paid (e.g., the property tax reference number).
- **Comments** are used to describe anything unusual about the non-CIS payment.

The contents of the **Payment Segments** section depends on the number of payment segments linked to the payment. If more than 25 payment segments exist, a message appears that allows you to navigate to Payment - Payment Segments where you can view the payment segments (this tab page has special functionality that allows you to defined which payment segments should be displayed). If 25 or fewer payment segments exist, the grid contains information about each payment segment:

- **Contract Information** contains basic information about the contract.
- **Distributed Amt** is the amount of the contract's debt relieved by the payment.

The following information appears at the bottom of the page:

- **Payment Amount** is the amount of the payment.
- **Distributed Amount** is the sum of the payment segments linked to the payment.
- The **Difference** between the **Payment Amount** and the **Distributed Amount** appears if it is non-zero.
- **Billed Amount** is the amount of the bill preceding the payment date.
- **Delinquent Amount** is the amount of the customer's debt that was due on / before the prior bill's due date.
- **Current Balance** is the account's current balance.

Refer to Payment Action Codes for information about the action buttons on this page.

Copyright © 2024, Oracle and/or its affiliates.

Payment - Pay Segments

You can use this page to view all or selected payment segments linked to a payment.

Open **Payments, Payment, Payment Segments** to view this information.

CAUTION: Most payments are maintained using the Payment Event page. The transaction described below is typically only used to view a payment's payment segments.

Note: If the payment has more than 25 payment segments, the search criteria are intentionally left blank in order to avoid retrieving all payment segments (with the resultant slow response times). You must therefore use the **Contract Filter** to define the type of payment segments that should be retrieved. See the `Description of page` below for more information about this page's search criteria.

Description of page

Payment Info contains a concatenation of important information about the payment.

Payment ID is the system-assigned unique identifier of the payment.

Payment Event ID is the system-assigned unique identifier of the payment event. The adjacent info contains a concatenation of the payment date, amount, and the name of the main customer on the account that remits the event's tender. If multiple tenders exist, the customer's name is not displayed. A warning may also be displayed. The possible warnings are `Unfrozen Payments` and `The Payment Event is unbalanced`.

Payment(s) contains the total number and value of payments linked to the payment event.

Tender(s) contains the total number and value of tenders linked to the payment event.

Note: If the payment has more than 25 payment segments, the search criteria are intentionally left blank in order to avoid retrieving all payment segments (with the resultant slow response times). You must therefore use the **Contract Filter** to define the type of payment segments that should be retrieved.

Use the **Contract Filter** to define the types of contracts whose payment segments appear in the grid. The following options are available:

- `All.` Use this option if you do not wish to restrict payment segments based on contract attributes.
- `Contract Type.` Use this option to restrict payment segments to those whose contracts are linked to a given **Division** and **Contract Type**.

Don't forget to click the search button after changing the **Contract Filter**.

The grid that follows contains the payment segments that match your search criteria. The following information appears in the grid:

- Click the go to button to transfer to view the financial transaction related to the payment segment. This is only available after the payment has been frozen.
- **Contract Information** contains summary information about the contract.
- **Distributed Amt** is the amount of the contract's debt relieved by the distribution.
- **Adjustment ID** is displayed if there is an adjustment associated with the payment. Refer to Loan Overpayments for an example of when an adjustment is associated with a payment.

The following information appears at the bottom of the page:

- **Payment Amount** is the amount of the payment.
- **Distributed Amount** is the sum of the payment segments linked to the payment.
- The **Difference** between the **Payment Amount** and the **Distributed Amount** appears if it is non-zero.
- **Billed Amount** is the amount of the bill preceding the payment date.
- **Delinquent Amount** is the amount of the customer's debt that was due on / before the prior bill's due date.
- **Current Balance** is the account's current balance.

Copyright © 2024, Oracle and/or its affiliates.

Payment - Manual Distribution

You can use this page to view contracts linked to the payment's account and to manually distribute a payment amongst specific contracts.

Open this page using **Payments, Payment, Manual Distribution**.

CAUTION: Most payments are maintained using the Payment Event page. The transaction described below is typically only used to view a payment's payment segments and to manually distribute a payment amongst an account's contracts.

Note: If the payment's account has more than 25 contracts, the search criteria are intentionally left blank in order to avoid retrieving all contracts (with the resultant slow response times). You must therefore use the **Contract Filter** to define the type of contracts that should be retrieved. See the `Description of page` below for more information about this page's search criteria.

Description of page

Payment Info contains a concatenation of important information about the payment.

Payment ID is the system-assigned unique identifier of the payment.

Note: If the payment's account has more than 25 contracts, the search criteria are intentionally left blank in order to avoid retrieving all contracts (with the resultant slow response times). You must therefore use the **Contract Filter** to define the type of contracts that should be retrieved.

Use the **Contract Filter** to define the types of contracts that appear in the grid. The following options are available:

- `All.` Use this option if you want to view all contracts.
- `Contract Type.` Use this option to restrict contracts to those linked to a given **Division** and **Contract Type**.

Don't forget to click the search button after changing the **Contract Filter**.

The grid that follows contains the contracts that match your search criteria. The following information appears in the grid:

- **Contract Information** contains summary information about the contract.
- **Distributed Amt** is the amount of the contract's debt relieved by the payment. This field is gray after the payment is frozen.
- **Billed Amt** is the amount that was billed for the contract on the bill preceding the payment date.
- **Delinquent Amt** is the amount of debt associated with financial transactions that appear on overdue bills.
- **Current Balance** is the amount currently owing for this contract.

Note: If the account being paid is an **open item** account, then an alternate grid is displayed. The alternate grid is designed to break down payable balances by individual bill and contract. Both unpaid bill balances and new charges are displayed. After distribution, a separate match event will be created for each bill paid or partially paid.

The open item manual distribution grid contains the following columns:

- **Bill Date**
  - Shows bill date if the row contains an account's unpaid bill.
  - Shows ' ***New Charges*** ' if the row contains an account's unbilled new charges.
  - Shows ' ***Payment Segment*** ' for the following conditions:
- Row is a pay segment of a cancelled payment.
- Row is a pay segment of an overpayment to the highest priority contract or an excess credit contract.
- Row is a pay segment of a balance forward payment distribution.
- **Bill ID** - Indicates the bill against which the payment is distributed.
- **Contract Information** contains summary information about the contract.
- **Distributed Amount**

Copyright © 2024, Oracle and/or its affiliates.

  - You can enter an amount to distribute if the payment is non-frozen and non-cancelled, otherwise, this field is protected.
- **Billed Amount**
  - The amount that was billed for the contract on the row's bill. If the row does not represent a bill amount then this will be zero.
- **Unpaid Amount**
  - This is determined by the "determine open-item bill amount" algorithm as defined on the installation option.
  - This column will contain zero if:
- The row is linked to an open non-disputed match event with other bills' FTs on it. In this case, it is not possible to determine the bill amount.
- The row is linked to an open disputed match event.

For either grid, the following information appears at the bottom of the page:

- **Payment Amount** is the amount of the payment.
- **Distributed Amount** is the sum of the payment segments linked to the payment.
- The **Difference** between the **Payment Amount** and the **Distributed Amount** appears if it is non-zero.
- **Billed Amount** is the amount of the bill preceding the payment date.
- **Delinquent Amount** is the amount of the customer's debt that was due on / before the prior bill's due date.
- **Current Balance** is the account's current balance.

Payment - Characteristics

To update payment characteristics, open **Payments, Payment** and navigate to the **Characteristics** tab.

Description of Page

The **Characteristics** collection contains information that describes miscellaneous information about the payment. The following fields display:

Characteristic Type Indicate the type of characteristic.

Sequence Controls the order in which characteristics of the same type are displayed.

Characteristic Value Indicate the value of the characteristic.

Note: You can edit the characteristics even when the payment is in the **Frozen** or **Cancelled** status.

Payment Action Codes

The topics that follow describe each of the actions that appear on the payment pages.

Distribute (A Payment)

Clicking the **Distribute** button causes a payment to be distributed amongst the account's contracts. Refer to Distributing A Payment Amongst An Account's Contracts for a description of how this is achieved.

This button is enabled if a payment is `Incomplete`, `Error`, or `Freezable`. Refer to Payment Lifecycle for more information about these status values.

If the payment is distributed appropriately, click the Freeze button to freeze the payment and its payment segments.

Redistribute (A Payment)

Clicking the **Redistribute** button causes a payment to be redistributed amongst the account's contracts. Refer to Distributing A Payment Amongst An Account's Contracts for a description of how this is achieved.

Copyright © 2024, Oracle and/or its affiliates.

This button is enabled if a payment is `Incomplete`, `Error`, or `Freezable`. Refer to Payment Lifecycle for more information about these status values.

If the payment is distributed appropriately, click the Freeze button to freeze the payment and its payment segments.

> **Note:** This button is only available on the Payment - Main and Payment - Manual Distribution pages.

Freeze (A Payment)

Clicking the **Freeze** button causes a payment and its payment segments to become `frozen`. Refer to Payment Lifecycle for more information about freezing.

This button is enabled if the payment is `Freezable`.

> **Note: If problems are detected after freezing.** A payment may not be changed after it is `frozen`. All subsequent changes must occur by canceling the frozen payment and creating a new one.

Cancel (A Payment)

Clicking the **Cancel** button causes the payment and its payment segments to be `canceled`. Canceling a payment (as opposed to canceling a tender) is typically only done if the original distribution was frozen and it's incorrect.

This button is enabled if the payment is `Frozen`. Refer to Payment Lifecycle for more information about these status values.

Transfer (A Payment)

The **Transfer** button is enabled if the payment is `Frozen`. Refer to Payment Lifecycle for more information about these status values.

If the payment being transferred was created using payment event distribution rule(s), the payment transfer should also be done using the distribution rules method. Hence, a different dialog opens when you click **Transfer** based on whether the payment's payment event is associated with at least one distribution detail or not.

Transfer Payment (No Distribution Details Exist)

Payment transfer causes the payment and its payment segments to be `canceled` and a new payment to be created. Transferring a payment is typically only done if the original distribution was targeted at the wrong account but all tender information for the pay event is correct. Unlike the payment event transfer, transferring a single payment does not affect the pay event's tender information.

You must specify the following parameters in the **Payment Transfer** window to transfer a payment:

- **Account ID** is the account to which the payment should be transferred.
- **Cancel Reason** defines why you are performing the transfer.
- **Match Type** and **Match Value** are defaulted based on their values for the payment event that you are transferring. If necessary, modify the values appropriately for the account to which you are transferring the payment event. Match Type and Match Value are used for an account that belongs to an open item customer class or to restrict the distribution of the payment to a specific contract. Refer to Payment Event - Main for more information.
- Turn on **Freeze Payment** if the transferred payment should be frozen automatically. You may want to leave this turned off if you want to examine the payment distribution before freezing the payment.
- **Non CIS Name**, **Reference Number**, and **Non CIS Comments** appear if the **Account ID** belongs to a customer class that is used for non-CIS payments. Refer to Setting Up Customer Classes for more information.

Clicking **OK** causes the following to take place:

- The original payment and its segments are cancelled.
- A new payment is added for the transfer to account. The payment characteristics associated with the original payment are copied to the new payment.
- The payment is distributed amongst the new account's contracts.

Copyright © 2024, Oracle and/or its affiliates.

- If so designated, the new payment is frozen.

Transfer Payment (Distribution Details Exist)

Payment transfer causes the payment to be `canceled` and new payment(s) to be created using new payment event distribution rule(s). Transferring a payment is typically done when a payment event has multiple payments and one or more payments have been assigned to the wrong contract. Each of the incorrectly directed payments can be transferred to the correct contract(s) using the **Transfer via Distribution Rules** dialog.

Total Amount is the total payment amount to transfer from.

You must specify the following parameters in order to transfer a payment:

- **Distribution Rule** is the contract to which the payment should be transferred.
- **Rule Value** is the value associated with the payment and expected by the distribution rule.
- **Amount** is the payment amount.
- **Cancel Reason** defines why you are performing the transfer.

Clicking **OK** causes the following to take place:

- The payment associated with the original account is canceled.
- The new set of distribution details is processed, creating new payment(s) for the transfer-to contract(s). The payment characteristics associated with the original payment are copied to the new payment.
- The new set of distribution details is added to the payment event. The original distribution details are kept.

Note: Distribution rule entries can have a zero amount. Refer to Rule Value Can Capture Additional Information for more information about how to use distribution details to capture additional payment related information.

Note: Determine Tender Account. This dialog does not call the `Determine Tender Account` algorithm defined on distribution rule as this action does not rebuild the payment tender collection.

Delete (A Payment)

The **Delete** button deletes a payment. This button is enabled if the payment is `Incomplete`, `Error`, or `Freezable`.

Payment Portal

The base package is shipped with a configured payment portal that can be used for payment processing. The portal supports the following functionality that is not supported in the base Payment Event - Add Dialog:

- Processing of payments that require authorization (i.e. credit card payments)
- Creating non CIS payments based on payment templates

Fastpath: Refer to The Big Picture of Portals and Zones for a description of portal and zone functionality.

Credit Card Payments

The payment portal provides the functionality needed for posting and authorizing customers' credit card charges in real-time. If the credit card authorization is successful, the payment will be added automatically (assuming you selected the appropriate payment freeze action via the portal).

Fastpath: Refer to How To for instructions describing how to add a payment that requires authorization.

Copyright © 2024, Oracle and/or its affiliates.

Non CIS Payments

The payment portal allows you to add non-CIS payments using a payment template. Payment Templates can be configured for common types of payment allocations to pre-populate the payment distribution, allowing non-CIS payments to be directly allocated against the appropriate distribution codes.

Fastpath: Refer to How To for instructions describing how to add a non-CIS payment using a payment template.

How To

The topics in this section describe how to perform common payment maintenance functions.

How To Add A New Payment Event

There are several ways to add a new event:

- You can use Payment QuickAdd to quickly add one or more payment events. You'd use this approach to add simple payments where no manual intervention is required. By "simple payment" we mean:
  - The account is both the tendering account and the account whose debt is being relieved by the payment
  - The payment date is the current date
  - The payment should be distributed amongst the account's contracts using standard distribution logic
- If applicable to your business practice, you can use Payment Event QuickAdd to quickly add one or more payment events using distribution rules.
- You can use Payment Event Maintenance to add a payment event. You would use this approach if multiple forms of payment are remitted (e.g., cash and a check) or if there are multiple payors and/or payees linked to the payment event.

Note: By default the system opens the Payment Event - Add Dialog when you navigate to this page in add mode. If you have opted to always use the payment event distribution rules method as your default method, the Payment Event QuickAdd (Single Payment Event) page appears instead. Refer to these pages for more information.

How To Cash A Check

If you allow customers to cash checks, you'll have a payment event with two tenders:

- The first tender contains information about the check.
- The second tender contains information about the cash refund (the tender amount is a negative amount equal to the amount of the check).

The interesting aspect of this payment event is that it has no payments because the sum of the tenders is zero.

We start this explanation in the middle of How To Add A New Payment Event.

In the **Payment Event Add** dialog, be sure to select `Do Not Distribute` before clicking **OK**.

- On the **Main** page, click **-** to remove the payment (cashing a check is a payment event with no payments).
- Transfer to the **Payment Event, Tenders** page.
- **Insert** a row in the tender scroll (click the + button) for the cash refund tender. In this row enter a tender type of "cash" and a tender amount of negative the check amount.
- Save the event (note there is nothing to distribute or freeze).

Copyright © 2024, Oracle and/or its affiliates.

How To Allocate The Tender Amount To Multiple Accounts

When you add a payment event, the system automatically creates a single payment for the account that remitted the funds. If someone is remitting funds for someone other than themselves, you must change/add payments. This section describes how to do this.

> **Note:** This section assumes you chose to add the payment event using the Payment Event - Add Dialog. Refer to How To Add A New Payment Event for the complete list of options.

In the Payment Event - Add Dialog, be sure to select `Do Not Distribute` before clicking **OK**.

Once on the payment event page, go to the Tenders tab and do the following:

- If the remitting account shouldn't have received any part of the payment, **Remove** it by clicking the - button. Alternatively, you can just change the account id to reflect the recipient.
- If multiple accounts receive the remitted funds, **Insert** one row in the payment scroll (click the + button) for each additional account.
- When the payment event is balanced, **Save** it. Then **Distribute** and **Freeze** the payment(s). Refer to Payment Event Actions for more information on these action buttons.

How To Print Receipts And Endorsements

The system can be configured to allow you to endorse checks and print receipts using special cashiering station printers. This section describes the print options available.

> **Note:** These options are only available if they have been installed in your system. Installing these options are a delivery and installation issue and are not within the domain of this document.

The print functions are available from the Payment Event, Payment QuickAdd and Payment Event QuickAdd transactions. Refer to those sections for more information.

The print dialog functions are described below.

Description of Page

Use the **Endorse** button to endorse checks using the cashier station printer. This is only enabled if one of the tender types indicates a check. If there were multiple checks, use the scroll buttons to select them. Feed the appropriate check in the printer and an endorsement message is printed on the back.

Use the **Receipt** button to print a receipt using the cashier station printer. Choose a **Short** or **Long** option. Use the **Duplicate** button to print a duplicate receipt - this prints the receipt with a "duplicate" label.

Use the **Stub** button to print special information in bill stub format (account, name, amount). Feed a slip of paper into the printer and the special stub information will be printed.

When you are finished, click **Done**; click **Cancel** to exit at any time.

> **Note:** The various text strings that are printed on the receipt and the endorsement are defined on Installation Options - Messages.

How To Cancel A Tender

If a tender is no longer valid (e.g., a check bounces), the tender must be canceled. The following steps explain how to do this.

> **CAUTION:** These steps only make sense in the context of the page used to maintain tenders. Refer to Payment Event - Tenders for the details.

Copyright © 2024, Oracle and/or its affiliates.

You navigate through the pages as follows:

1. Open the payment event - tenders page for the tender in question. Proceed to step 2.
2. Click **Cancel**. Proceed to step 3.
3. Before the system cancels the tender, you must specify the cancel reason. The tender cancel window opens.

In the tender cancel window, select a **Cancel Reason** to describe why the tender is being canceled. The system also needs to know which **Bank Account To Charge** the cancellation against. The system will default the **Bank Code** and **Bank Account** from the original bank information used when the tender was deposited. You can override them here.

When you click the **OK** button, the system cancels the tender and ALL payments linked to the event.

If the cancellation reason you supply is indicated as being "non-sufficient funds", the system will generate an adjustment to levy a NSF charge. Refer to Contract Type - Main Information for more information.

Note: When non-canceled tenders are still linked to the event. If there are multiple tenders linked to the payment event, you must either add new payment(s) that equal the sum of the event's non-canceled tenders or cancel the remaining tenders.

Note: Cancellation after cash back. If a customer tenders a check for $100, but only owes $80, the system will recommend returning $20 in cash to the customer (assuming the tender type for checks allows overpayment). If the tender type for checks is not "like cash", a second tender is created for -$20 (the first tender is for $100). If the check subsequently bounces, both tenders must be cancelled.

How To Transfer A Payment From One Account To Another

If you need to transfer a payment amount from one account to another, you can use the **Transfer** button either on the payment event page or the payment page.

Fastpath: Refer to Transferring A Payment for a description of when to use each option.

How To Distribute A Payment To A Specific Contract

When you click **Distribute**, the system uses the payment distribution algorithm defined on the account's customer class to allocate the payment amongst the account's contracts. In this section, we describe how to override this distribution (e.g., when you need to allocate a payment to specific contract(s)).

We start this explanation in the middle of How To Add A New Payment Event.

In the **Payment Event Add** dialog, be sure to select `Do Not Distribute` before clicking **OK**.

- **Save** the payment event (this also saves the payment and tender information).
- Use the **Payment Amount** context menu to transfer to the **Manual Distribution** page.
- Fill in the desired **Distributed Amt** for each contract and save the payment.
- Click **Freeze** to freeze the payment when the amount distributed is equal to the payment amount.

How To Get An Unbalanced Tender Control In Balance (Fixing Over/Under)

In order to balance a tender control that is out-of-balance, you must create a tender for an account associated with your company.

Note: The over / under account. Your organization must set up a company-use account with a contract whose contract type references the over/under distribution code. This account should be linked to the person ID associated with company usage.

Copyright © 2024, Oracle and/or its affiliates.

- If you have more money in the drawer than you have tenders, then you're "over". In this situation, you need to record a payment for the "over" amount. This will cause cash to be debited by the over amount, and the expense account associated with the account's contract to be credited.
- If you have less money in the drawer than you have tenders, then you're "under". In this situation, you need to record a negative payment for the "under" amount. This will cause cash to be credited by the over amount, and the expense account associated with the account's contract to be debited.

Note: Important! You must determine the over/under amount for each tender type and then enter the respective over/under tender for the company-use account. For example, if you have more cash but fewer checks then you must enter two tenders for the company-use account (but you can do this on the same event). Check out the following table for an example:

Tender Type	Starting Balance	Tenders Received	Turn Ins	Actual Ending Balance (Entered by Cashier)	Expected Ending Balance	Over/Under Amount
Cash	$150.50	$5,000	$4,000	$1151	$1150.50	Over $0.50
Check	-	$1,000	$750	$249	$250	Under $1

In this example, you'd have to create two tenders for the company-use account:

- Tender type: Cash, Tender amount: +$0.50
- Tender type: Check, Tender amount: -$1.00

The sum of the tenders is -$0.50 and this is the amount that will be distributed to the company-use account's contract (debit over/under expense, credit cash).

Note: You must reopen the tender control. Before you can add an over/under tender to a tender control, you must re `open` the tender control (tenders may only be added to `open` tender controls).

How To Add A Non-CIS Payment Using A Template

The payment portal allows you to add non-CIS payments using a payment template. Payment Templates can be configured for common types of payment allocations to pre-populate the payment distribution and allowing non-CIS payments to be directly allocated against distribution codes. The following steps explain how to do this.

You navigate through the pages as follows:

- From the main menu open **Payments, Payment Portal**.
- The **Add Non CIS Payment Zone** shows a list of all active payment templates. Use the **Add** button to add a non-CIS payment for a specific payment template.
- Enter the payment and tender details (**Payment Amount**, **Payment Date** defaults to the current date, **Tender Source** is used to find an open tender control when adding the tender).

Note: Authorizing Tenders. If the **Tender Type** selected requires authorization (i.e. credit card payments), additional authorization details are required (e.g. credit card type, credit card number, etc.). Only one tender that requires authorization can exist for a single payment event.

Fastpath: For information about tender types requiring authorization, refer to How To Authorize A Credit Card Tender.

- Enter the non-CIS information for the customer remitting the payment (**Customer Name**, **Agreement Number** and **Comments** to describe anything unusual about the non CIS payment).

Note: Payor Account ID. The Non CIS Account Id specified on the payment template is used as the tendering account.

Copyright © 2024, Oracle and/or its affiliates.

> **Note: Including a CIS payment.** If a portion of the tender is for a CIS payment, ensure that the **Add CIS Payment** checkbox is selected. The system will process the non-CIS payment, and then navigate the user to the Payment Event page where the CIS payment can be included.

- The distribution details are defaulted from the payment template but may be overridden. You can add additional distribution lines. Enter the payment amount and Reference Number for each applicable distribution line.
- When you click the **OK** button, the system creates the payment event and tender(s). A single payment is created for each non-zero amount entry in the distribution details list. The payment template is captured as a characteristic on the tender. The corresponding distribution code, GL account and reference number are captured as characteristics on each payment. When the financial transactions are created for these payments, the distribution code (and GL account) is retrieved from the payment characteristic collection.

How To Authorize A Credit Card Tender

You can use the Payment Portal to add a payment event for a tender that requires authorization. The following steps explain how to do this.

If your organization is using the base tender authorization functionality, you can navigate through the pages as follows:

- From the main menu open **Payments, Payment Portal**.
- The **Add CIS Payment Z one** only expands if there's an account in context. To expand this zone, either search for an account using Control Central or use the Tender Search available on the **Payment Portal**.
- Use the **Add Payment** button to add a payment for the account displayed in the zone.
- The information displayed is the same as that on the Payment Event - Add Dialog, except for the following:
- • Use **Distribute Action** to describe what you'd like to have happen when you click the **OK** button. The dropdown contains the following values - `Distribute and Freeze if OK`, `Manual Distribution` and `Manual Distribution - Multiple Payments`

> **Note:** `Note: The Distribute and Freeze if OK and Manual Distribution options are also present on the Distribute Action on the payment event add dialogue screen. The third option Manual Distribution - Multiple Payments is special for credit card authorization processing. This option is similar to the Manual Distribution option except that the user is navigated to the Payment Event maintenance page after authorization has taken place.`

- If the **Tender Type** selected requires authorization (i.e. credit card payments), additional authorization details are required (e.g. credit card type, credit card number, etc). This information is displayed in the **Credit Card Information** section.

> **Note: Authorizing Tenders.** Only one tender that requires authorization can exist for a single payment event.

- The **Authorization Override** checkbox is only displayed if the tender type selected requires authorization. Use the **Authorization Override** checkbox to indicate whether or not tender authorization should take place. If **Authorization Override** is checked, an authorization code must be entered and the system bypasses the authorization web service call.
- Click **OK** to save the payment and navigate to the appropriate page.
- Assuming the credit card is accepted, the authorization code returned by the credit card authorization provider is captured as a tender characteristic.
- If the credit card is declined, the tender will not be added.

How To Reverse A Credit Card Tender

The steps required to reverse a credit card tender are the same as the steps for How To Cancel A Tender.

The reversal code from the credit card authorization service is stored as a tender characteristic.

Copyright © 2024, Oracle and/or its affiliates.

Fastpath: For more information on authorizing credit card payments, refer to the following business objects: `Tender Type - Credit Card with Authorization`, `CyberSource - Credit Card Authorization` and `CyberSource - Credit Card Reversal`.

How to Distribute Overpayment on the Next Bill Completion

If the overpayment is made by an account, the excess credit is transferred to account's excess credit contract. The system allows you to distribute credit from the account's excess credit contract against the next bill during the post bill completion. To enable this functionality, you need to create an algorithm using the **CR-DSTRBTN** algorithm type and attach it to the **Post Bill Completion** algorithm spot of the excess credit contract's contract type. This algorithm will create debit and credit adjustments. A debit adjustment is created against the excess credit contract. One or more credit adjustments are created against the bill.

While creating an algorithm using the **CR-DSTRBTN** algorithm type, you need to specify the following parameters:

- **Debit Adjustment Type** - Used to specify the adjustment type using which you want to create debit adjustment against the excess credit contract.
- **Credit Adjustment Type** - Used to specify the adjustment type using which you want to create credit adjustments against the bill.
- **Match Type** - Used to specify the match type whose payment distribution algorithm you want to use for distributing credit from the excess credit contract against the bill. Note that the **Entity** flag of the match type must be set to **Bill**.
- **Adjustment Characteristic Type** - Used to specify the characteristic type. You must specify a characteristic type where the type of char value is set to **Foreign Key Value** and characteristic entity is set to **Adjustment**. While defining the credit adjustment type, this characteristic type must be added in the **Adjustment Characteristics** tab. It indicates that this characteristic must be defined for adjustments which are created using the credit adjustment type. It is used to link debit adjustment (created against the excess credit contract) with one or more credit adjustments (created against the bill).

Note: If a bill has a credit line item, a debit adjustment is created to offset the credit line item before distributing the excess credit against the bill. This debit adjustment is created using the adjustment type specified in the **Credit Adjustment Type** parameter. This debit adjustment is also linked to the debit adjustment which is created against the excess credit contract.

All parameters in this algorithm type are mandatory.

Financial Transactions On A Payment

Open **Report and Inquiry, Financial Transactions on Payment** to view the financial transactions on a payment.

Note: You can also open this page using the go to buttons that prefix the financial transaction summaries on Payment - Main.

Description of page

Payment Id is the system-assigned unique identifier of the payment.

Account ID is the payment's account.

The area beneath **Account ID** provides you with options that control which financial transactions appear in the grid. The following points describe the various options:

- Use the **Contract Filter** to define the types of contracts whose financial transactions appear in the grid. The following options are available:
-
  - `All.` Use this option if you do not wish to restrict financial transactions based on contract attributes.
  - `Contract ID`. Use this option to restrict financial transactions to those of a specific contract.

Copyright © 2024, Oracle and/or its affiliates.

  - `Contract Type`. Use this option to restrict financial transactions to those whose contracts are linked to a given **Division** and **Contract Type**.
- Use **Match Event Status Filter** to restrict the financial transactions based on the status of their match event. This filter only appears if the payment's account is an open-item customer. The following options are available:
- - `All`. This option shows all financial transactions.
  - `Balanced`. This option shows all financial transactions whose match event is `balanced`.
  - `Disputed`. This option shows all financial transactions whose match event is `disputed`.
  - `Unbalanced`. This option shows all financial transactions whose match event is `unbalanced`.
  - `Unmatched`. This option shows all financial transactions that are not linked to a match event.

CAUTION: Don't forget to click the search button after changing the filters or after selecting a new Payment Id.

The grid that follows contains the financial transactions that match your search criteria. The following information is displayed:

- **Match Event Status** shows the status of the financial transaction's match event. This column only appears if the account is an open-item customer.
- **FT Type** displays the type of financial transaction. Click on the hyperlink to transfer to Financial Transaction - Main. On this page, you can change certain aspects of the FT in question.
- **Accounting Date** is the date the system uses to determine the financial transaction's accounting period in your general ledger.
- **Current Amount** contains the financial transaction's effect on the contract's current balance.
- **Payoff Amount** contains the financial transaction's effect on the contract's payoff balance. The **Payoff Amount** will be dim if it equals the **Current Amount**.
- **Show on Bill** indicates if information about the financial transaction appears on the customer's bill.
- **Contract Information** contains a summary of the respective contract.
- **Financial Transaction ID** is the system-assigned unique identifier of the financial transaction.

At the bottom of the page is a summary of the financial transactions that match the search criteria.

Payment History

The **Payment History** screen allows you to search for payments using various search criteria. This screen consists of the following zones:

- Search on page 376

Search

The **Search** zone allows you to search for payments using various search criteria. This zone contains the following two sections:

- **Search Criteria** - The **Search Criteria** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Division	Used to search for payments that belong to a particular division. **Note:** You can only view those divisions to which you have access in the **Division** list.	Yes

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
Payee Account ID	Used to indicate the account for which the payment was made. You can search for an account by clicking the **Search** (🔍) icon corresponding to the field.	No
Payee Name	Used to indicate the person for whom the payment was made.	No
Payor Account ID	Used to indicate the account which has made the payment. You can search for an account by clicking the **Search** (🔍) icon corresponding to the field.	No
Payor Name	Used to indicate the person who has made the payment.	No
Payment Date From	Used to search for payments made from a particular date onwards.	No
Payment Date To	Used to search for payments made till a particular date.	No
Payment Amount From	Used to search for payments whose amount is greater than or equal to a particular amount.	No
Payment Amount To	Used to search for payments whose amount is less than or equal to a particular amount.	No
Tender Type	Used to search for payments made through a particular tender type.	No
Tender Source	Used to search for payments made through a particular tender source. You can search for a tender source by clicking the **Search** (🔍) icon corresponding to the field.	No
Check Number	Used to specify the check number through which the payment was made.	No
Match Type	Used to search for payments which are matched using a particular match type.	Yes (Conditional) **Note:** If you enter the match value as a search criteria, you have to select the match type.
Match Value	Used to indicate the entity (such as bill, contract, and so on) against which the payment is matched.	Yes (Conditional) **Note:** If you specify the match type as a search criteria, you have to enter the match value.

- **Search Results** - On clicking the **Search** button, the search results appear based on the specified search criteria. The **Search Results** section contains the following columns:

Column Name	Column Description
Account Information	Displays information about the account. In addition, this column has a context menu which helps in navigating to other screens in the application. **Note:** It has a link. On clicking the link, the **Account** screen appears where you can view the details of the respective account.
Payment Information	Displays information about the payment. In addition, this column has a context menu which helps in navigating to other screens in the application.

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
	Note: It has a link. On clicking the link, the **Payment** screen appears where you can view the details of the respective payment.
Payment Date	Displays the date when the payment was made.
Payment Currency	Indicates the currency in which the payment was made.
Payment Amount	Displays the amount paid through the tender.
Payment Status	Indicates the current status of the payment. The valid values are: • Incomplete • Error • Freezable • Frozen • Canceled
Tender Type	Displays the type of tender through which the payment was made.
Tender Amount	Displays the tender amount.
Check Number	Displays the check number if the payment was made through check.
Match Type	Indicates the match type using which the payment was matched.
Match Value	Indicates the entity (such as bill, contract, and so on) against which the payment was matched.
GL Transfer Status	Indicates the current status of GL transfer for the payment. The valid values are: • Generated • Distributed
GL Transfer Date	Displays the date when the GL transfer was done for the payment.

Note: Along with division, you must specify at least one search criterion while searching for a payment.

Related Topics

For more information on...	See...
How to search for a payment	Searching for a Payment on page 378

Searching for a Payment

Prerequisites

To search for a payment, you should have:

- Divisions, tender types, tender sources, and match types defined in the application

Procedure

To search for a payment:

1. Click the **Menu** link in the **Application** toolbar.
   A list appears.
2. From the **Main** menu, select **Report and Inquiry** and then click **Payment History**.

Copyright © 2024, Oracle and/or its affiliates.

The **Payment History** screen appears.

3. Enter the search criteria in the **Search** zone.

> **Note:** ORMB search engine supports wildcard search, where you can substitute the percentage (%) symbol as a stand in for any word or letter in a search criteria. You can use the '%' wildcard character in all input fields except the date and ID fields. The '%' wildcard character is suffixed automatically at the end of the partial search criteria. Therefore, you may or may not specify the wildcard character at the end of the partial search criteria. However, you have to prefix the wildcard character manually wherever required.

4. Click **Search**.
   A list of payments that meet the search criteria appears in the **Search Results** section.
5. If required, you can export the list of payments in the Excel format by clicking the **Export To Excel** menu option from the **Explorer Zone** (▤) menu.

Related Topics

For more information on...	See...
Payment History screen	Payment History on page 376
Search zone	Search on page 376

Remittance Summary

The **Remittance Summary** screen allows you to view a list of payments remitted by a bank using a tender source. It also displays consolidated remittance report for each bank and tender source combination. This screen consists of the following zones:

- Search on page 379
- Payments on page 380

Search

The **Search** zone allows you to search for all payments remitted by a bank using a tender source. This zone contains the following two sections:

- **Search Criteria** - The **Search Criteria** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Tender Source	Used to specify the tender source.	No
Bank	Used to specify the bank code.	No
Payment Date From	Used when you want to search for payments which are made from a particular date onwards.	No
Payment Date To	Used when you want to search for payments which are made till a particular date.	No

> **Note:** You must specify at least one search criterion while searching for payments remitted by a bank.

- **Search Results** - On clicking the **Search** button, the search results appear based on the specified search criteria. The **Search Results** section contains the following columns:

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
Tender Source	Displays the tender source.
Bank	Displays the bank code. **Note:** It has a link. On clicking the link, the **Bank** screen appears with the details of the respective bank.
Total Amount	Displays the sum of total payment amount in suspense, total overpayment amount, and total frozen payment amount.
Total Payment Amount in Suspense	Displays the total amount of payments which are made on the suspense account.
Total Overpayment Amount	Displays the total amount of excess credit (overpayment) made for the account.
Total Frozen Payment Amount	Displays the total amount of payments which are in the **Frozen** status.

Related Topics

For more information on...	See...
How to search for payments remitted by a bank	Searching for Payments Remitted by a Bank on page 381

Payments

The **Payments** zone lists the payments which are remitted by the bank using the tender source. You can filter the payments, if required. This zone contains the following two sections:

- **Search Criteria** - The **Search Criteria** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Comments	Used when you want to filter the payments. The valid values are: • **Frozen Payment** - Used when you want to only view the payments which are in the **Frozen** status. • **Overpayment** - Used when you want to only view overpayments remitted by the bank using the tender source. • **Payment in Suspense Account** - Used when you want to only view the payments which are made on the suspense account.	No

- **Search Results** - On clicking the **Refresh** button, the search results are filtered based on the specified search criteria. The **Search Results** section contains the following columns:

Column Name	Column Description
Payment Date	Displays the date when the payment was made.
Tender Type	Displays the type of tender through which the payment is made.
Check Number	Displays the check number if the payment is made through check.

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
Payment Amount	Displays the amount paid using the tender source.
Payment ID	Displays the payment ID. **Note:** It has a link. On clicking the link, the **Payment** screen appears with the details of the respective payment.
Account ID	Displays the account for which the payment is made. **Note:** It has a link. On clicking the link, the **Account** screen appears where you can view the details of the respective account.
Comments	Displays additional information about the payment. The valid values are: • **Frozen Payment** - Indicates that the payment is in the **Frozen** status. • **Overpayment** - Indicates that overpayment is made for the account. • **Payment in Suspense Account** - Indicates that the payment is made on the suspense account.

By default, the **Payments** zone does not appear in the **Remittance Summary** screen. It appears when you click the **Broadcast** (ᯤ) icon corresponding to the bank and tender source combination in the **Search** zone.

Related Topics

For more information on...	See...
How to search for payments remitted by a bank	Searching for Payments Remitted by a Bank on page 381

Searching for Payments Remitted by a Bank

Prerequisites

To search for payments remitted by a bank, you should have:

- Tender sources and banks defined in the application

Procedure

To search for payments remitted by a bank:

1. Click the **Menu** link in the **Application** toolbar.
   A list appears.
2. From the **Main** menu, select **Report and Inquiry** and then click **Remittance Summary**.
   The **Remittance Summary** screen appears.
3. Enter the search criteria in the **Search** zone.

   Note: ORMB search engine supports wildcard search, where you can substitute the percentage (%) symbol as a stand in for any word or letter in a search criteria. You can use the '%' wildcard character in all input fields except the date and ID fields. The '%' wildcard character is suffixed automatically at the end of the partial search criteria. Therefore, you may or may not specify the wildcard character at the end of the partial search criteria. However, you have to prefix the wildcard character manually wherever required.

4. Click **Search**.
   A consolidated remittance report appears for the bank and tender source combination in the **Search Results** section.
5. Click the **Broadcast** (ᯤ) icon corresponding to the bank and tender source combination whose payment details you want to view.

Copyright © 2024, Oracle and/or its affiliates.

A list of payments remitted by the bank using the tender source appears in the **Payments** zone.

6. If required, you can export the consolidated remittance report and a list of payments in the Excel format by clicking the **Export To Excel** menu option from the **Explorer Zone** (▤) menu in the respective zone.

Related Topics

For more information on...	See...
Remittance Summary screen	Remittance Summary on page 379
Search zone	Search on page 379
Payments zone	Payments on page 380

Account Payment History

The payment history page shows all payments that have been distributed to an account's contracts. Open this page using **Report and Inquiry, Account Payment History**.

Description of Page

One row is displayed for every payment that has been distributed to an account's contracts. The payment date, payment amount, payment status and the tender source associated with the tender's tender control are displayed for each payment.

Note: Tender Source typically contains the description of the cash drawer in which the payment was made or the remittance processor that processed the payment. **Tender source** will be blank for automatic payments until they are interfaced to the financial institution. Refer to Downloading Automatic Payments for more information about interfacing automatic payments.

If you need to see more detailed information about the payment, click the Go To button to transfer to the payment event page.

Payment Tender History

The **Payment Tender History** page shows a list of all tenders linked to an open tender control for the current user. Open this page using **Report and Inquiry, Payment Tender History**.

Description of Page

One row is displayed for every tender that is linked to an open tender control to which the current user has access. The freeze date/time, payor account id, name, tender amount, tender type, pay tender id, tender status and payment date are displayed for each tender.

Payment / Tender Search

This page allows you to look for payments and/or tenders using a combination of search criteria. Open this page using **Payments, Payment / Tender Search**.

Description of Page

The top half of the page is where you enter the criteria used to search for payments and tenders.

Copyright © 2024, Oracle and/or its affiliates.

Note: Multiple search criteria may be specified. You can search for payments and tenders using a combination of search criteria. For example, if you enter both a **Payment Account** name of `Brandon` and a **Payment Amount** between `150` and `170`; only those payments for customers named `Brandon` whose amount is between `150` and `170` will be displayed.

CAUTION: Try to be as specific as possible when entering search criteria. Why? Because entering open-ended search criteria may have a severe impact on response times. For example, if you know the payments you're looking for have a payment date sometime in `January 2003` and the customer's name is `Brazil,John`, then enter both of these criteria. If you also know the payment amount is between `150` and `170`, enter these values too.

The following table describes each of the different search methods.

Search Method	Description
Search for	Use this field to define if you're searching for `Payments`, `Tenders` or both `Payments and Tenders`. The value entered in this field controls which of the remaining search methods are enabled.
Distribution Rule	Use this field if you're searching for a tender or a payment and know the distribution rule and value used to created it. This section appears only if you have configured your system to allow the payment event distribution rules method to be used.
Payment Account	Use this field if you're searching for a payment and know the account ID or the name of any person linked to the account whose debt is relieved by the payment. - If you know the payment's account ID, first choose `Account` in the adjacent dropdown and then and enter the account's ID. You must enter all of the account ID. - If you know the name of any person linked to the account, first choose `Person Name` in the adjacent dropdown and then enter the person's name. You can enter all or part of the person's name (the more you enter, the faster the search will be). The name search is not case sensitive. A search method of `Person Name` defaults. If you leave the person name or ID blank, the system ignores this search method. These fields are protected if you've selected a **Search for** of `Tender`.
Payment Amount	Use this field if you're searching for a payment and know its amount. - If you know the exact amount, first choose `Equal To` in the adjacent dropdown and then enter the amount. - If you know the payment amount is between a given range of values, first choose `Between` and then enter the range of values. A search method of `Between` defaults. If you leave the amount(s) blank, the system ignores this search method. These fields are protected if you've selected a **Search for** of `Tender`.
Payor Account	Use this field if you're searching for a tender and know the account ID or the name of any person linked to the tendering account. - If you know the tender's account ID, first choose `Account` in the adjacent dropdown and then and enter the account's ID. You must enter all of the account ID. - If you know the name of any person linked to the tendering account, first choose `Person Name` in the adjacent dropdown and then enter the person's name. You can enter all or part

Copyright © 2024, Oracle and/or its affiliates.

Search Method	Description
	of the person's name (the more you enter, the faster the search will be). The name search is not case sensitive. A search method of `Person Name` defaults. If you leave the person name or ID blank, the system ignores this search method. These fields are protected if you've selected a **Search for** of `Payment`.
Tender Amount	Use this field if you're searching for a tender and know its amount. - If you know the exact amount, first choose `Equal To` in the adjacent dropdown and then enter the amount. - If you know the tender amount is between a given range of values, first choose `Between` and then enter the range of values. A search method of `Between` defaults. If you leave the amount(s) blank, the system ignores this search method. These fields are protected if you've selected a **Search for** of `Payment`.
Tender Source	If you're searching for a tender and you know the source of the tender (e.g., the lockbox, cashier, etc.), enter the tender source code. A search method of `Equal To` defaults. If you leave the tender source blank, the system ignores this search method. These fields are protected if you've selected a **Search for** of `Payment`.
MICR ID	Use this field if you're searching for a tender and know its MICR (magnetic ink character recognition) ID. - If you know the entire MICR ID, first choose `Equal To` in the adjacent dropdown and then and enter the entire MICR ID. - If you know the first X characters of the MICR ID, first choose `Like` in the adjacent dropdown and then and enter the entire MICR ID. A search method of `Like` defaults. If you leave the MICR ID blank, the system ignores this search method. These fields are protected if you've selected a **Search for** of `Payment`.
Payment Date	Use this field if you're searching for a tender or a payment and know its date. - If you know the exact date, first choose `Equal To` in the adjacent dropdown and then enter the date. - If you know the date is between a given range of values, first choose `Between` and then enter the date range. A search method of `Between` defaults. If you leave the date(s) blank, the system ignores this search method.

The system shows the total number of payments / tenders that satisfy your search results immediately below the grid.

The first group of payments / tenders is displayed in the grid at the bottom of the page. Different columns appear in the grid depending on the value of **Search for**.

- If you use a **Search for** of `Payment`, only payment-oriented columns appear in the grid.

Copyright © 2024, Oracle and/or its affiliates.

- If you use a **Search for** of `Tender`, only tender-oriented columns appear in the grid.
- If you use a **Search for** of `Payment and Tender`, both payment-oriented and tender-oriented columns appear in the grid.

CAUTION: If you use the `Payment and Tender` search method, and the resultant tenders / payments are linked to payment events that have multiple payments or tenders, multiple rows may be displayed for the payment event's tenders and payments. For example, if a tender with multiple payments is selected, a separate row will be displayed for every payment that matches your payment search criteria. If you don't specify any payment search criteria, a row will be displayed for every payment linked to the tender.

Payment Date This contains the date of the payment event associated with the payment / tender. This column appears regardless of the value of **Search for**.

Payment Account Info This contains a concatenation of important information about the account whose debt was relieved by the payment. This column does not appear if you use a **Search for** of `Tender`.

Payment Amount This contains the amount of the payment. This column does not appear if you use a **Search for** of `Tender`.

Payment Status This contains the status of the payment. Refer to Payment Lifecycle for more information. This column does not appear if you use a **Search for** of `Tender`.

Payment ID This contains the unique identifier of the payment. This column does not appear if you use a **Search for** of `Tender`.

Payor Account Info This contains a concatenation of important information about the account who tendered the payment. This column does not appear if you use a **Search for** of `Payment`.

Tender Amount This contains the amount of the tender. This column does not appear if you use a **Search for** of `Payment`.

Tender Source This contains the source of the tender of the tender. This column does not appear if you use a **Search for** of `Payment`.

MICR ID This contains the MICR (magnetic ink character recognition) ID of the tender. This column does not appear if you use a **Search for** of `Payment`.

Tender Status This contains the status of the payment. Refer to Tender Lifecycle for more information. This column does not appear if you use a **Search for** of `Payment`.

Pay Tender ID This contains the unique identifier of the tender. This column does not appear if you use a **Search for** of `Payment`.

Pay Event ID This contains the unique identifier of the payment event.

Payment Event Exception

Unbalanced payment events cause a record to be written to the payment event exception table with a message indicating the nature of the error.

To view the messages associated with the exception records, schedule the TD-UNBAL background process. This process generates a To Do entry for every record in the payment event exception table.

Fastpath: Refer to Unbalanced Payment Events for instructions describing how to correct an unbalanced payment event.

Copyright © 2024, Oracle and/or its affiliates.

Payment Exception

When the system is unable to distribute a payment, a record to be written to the payment exception table with a message indicating the nature of the error.

To view the messages associated with the exception records, schedule the TD-PYERR background process. This process generates a To Do entry for every record in the payment exception table.

After correcting the cause of the error, drill into the Payment and attempt to redistribute it.

Maintaining Deposit Controls

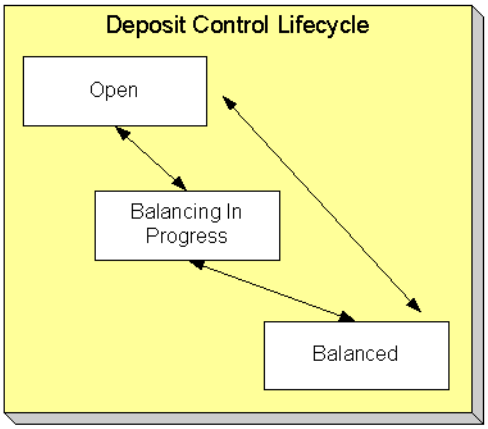
Deposit controls exist to give you administrative control over your cash drawers (and all other tender sources) and the subsequent deposit of funds at banks.

Fastpath: Refer to Tender Management and Workstation Cashiering for background information.

Note: The system creates most deposit controls behind-the-scenes. Most deposit controls are created by the system when it processes tenders from your remittance processor and lock boxes. You should only have to access the deposit control pages if you record payments in cash drawers. For information about how the system creates deposit controls, refer to Managing Payments Interfaced From External Sources. Also note that the automatic payment activation process also creates tender and deposit controls. Refer to Activating Automatic Payments for more information.

The Lifecycle Of A Deposit Control

The following diagram shows the possible lifecycle of a deposit control.



`Open` A deposit control is initially created in the `Open` state. While in this state, you may add new deposits to it, change the deposit amount on its deposits, and transfer tender controls into and out of it.

`Balancing In Progress` You change a deposit control's status to `Balancing In Progress` when you're ready to balance its contents. While in this state, you can change the deposit amount on its deposits and transfer tender controls out of it. If you need to add new deposits to it or transfer tender controls into it you must return it to the `Open` state.

`Balanced` You change a deposit control's status to `Balanced` when the sum of its tender controls is consistent with the total of its deposits. While in this state, you cannot modify its deposits or its tender controls. If you need to make modifications, you must return it to the `Open` or `Balancing In Progress` state.

Copyright © 2024, Oracle and/or its affiliates.

> **Note: Background processes and state transition.** When payments are interfaced from external sources, the system automatically creates a deposit control and links a tender control to it. When all payments have been successfully loaded, the system changes the state of the respective deposit control to `Balanced`.

Deposit Control - Main

The Main page contains core deposit control information. Open this page using **Payments, Deposit Control** and then navigate to the **Main** tab.

Description of Page

Deposit Control contains a concatenation of the deposit control's creation date, tender source type, and status.

Deposit Control ID is the system-assigned, unique identifier of the deposit control.

User is the person who created the deposit control.

Create Date/Time is the date and time the deposit control was created.

Tender Source Type is the type of tender control that has been linked to the deposit control. Valid values are: `Ad hoc`, `Auto Pay`, `Online Cashiering` and `Lockbox`. The system uses this information to prevent tender controls from different sources from being included under the same deposit control. In other words, you can't mix automatic payment, cashiering and lockbox tenders under the same deposit control.

Currency Code is the currency in which the deposit control's tenders are denominated.

> **Note:** The currency code defaults from the installation record.

The summary information that follows contains a summary of the starting balance and the tenders that are linked to the tender control.

- **Starting Balance** is the sum of the starting balances from all tender controls linked to the deposit control.
- **Total Tenders Amount** is the sum of tenders from all tender controls linked to the deposit control.
- **Total Tender Controls** is the number and amount of tender controls linked to the deposit control.
- **Total Tender Deposits** is the number and amount of tender deposits linked to the deposit control.
- **Expected Ending Balance** is the **Total Tender Control** minus **Total Tender Deposits**.
- **Ending Balance** is the actual amount of money in the tender control. This amount must equal the **Expected Ending Balance** before the tender control can be marked as `Balanced`.
- **Outstanding Over/Under** is **Ending Balance** minus **Expected Ending Balance**.

Deposit Control Status shows the status of the deposit control. Valid values are `Open`, `Balanced`, and `Balancing In Progress`.

For more information, refer to The Lifecycle Of A Deposit Control.

The **Balanced User ID** and **Balanced Date/Time** are populated when the status is changed to `Balanced`.

Use **Comments** to describing anything unusual about the deposit control.

Deposit Control - Tender Control

The Tender Control page contains a row for every tender control linked to the deposit control. Open this page using **Payments, Deposit Control, Tender Control**.

Description of Page

Deposit Control contains a concatenation of the deposit control's creation date, tender source type, and status.

Deposit Control ID is the system-assigned, unique identifier of the deposit control.

Copyright © 2024, Oracle and/or its affiliates.

The grid that follows contains a row for every tender control linked to the deposit control. No information about the tender controls may be modified on this page. To view or modify tender control information, click the Go To button.

Deposit Control - Tender Deposit

The Tender Deposit tab page contains a row for every tender deposit linked to the deposit control. Open this page using **Payments, Deposit Control** and navigate to the **Tender Deposit** tab.

Description of Page

Deposit Control contains a concatenation of the deposit control's creation date, tender source type, and status.

Deposit Control ID is the system-assigned, unique identifier of the deposit control.

The **Tender Deposit** scroll that follows contains one row for every financial institution into which the tenders will be deposited. To insert a new row, click the + button and fill in the following fields:

- **Tender Deposit ID** is maintained by the system. When a deposit is being created, there is no ID number displayed. Once a deposit is entered and saved, the system generates an ID and displays it here.
- **Deposit Amount** contains the amount to be deposited at the bank. **Currency Code** is the currency in which the deposit is denominated.
- **Reference ID** contains the ID of the deposit (if any).
- Use **Bank Code** and **Bank Account** to define where the tenders will be deposited.

Deposit Control - Turn Ins

The Turn Ins page contains a row for every turn-in event linked to the deposit control. Open this page using **Payments, Deposit Control** and navigate to the **Turn Ins** tab.

Fastpath: Refer to Turn Ins **for background information**.

Description of Page

Deposit Control contains a concatenation of the deposit control's creation date, tender source type, and status.

Deposit Control ID is the system-assigned, unique identifier of the deposit control.

The turn-ins grid contains one row for every turn-in event associated with the deposit control. A turn-in event is created when moneys originally deposited in respect of a tender control are turned in to the tender control's deposit control. Turn-in events are created and maintained on the Tender Control - Turn Ins page. Refer to Turn Ins for background information. The following information is displayed in the grid:

- **Turn In Status** defines if it is a new turn in `Awaiting approval` or has been `Approved` by the operator who is responsible for the Deposit Control. Once the turn-in is `Approved`, this field becomes display-only.
- **Tender Type** is the type of tender that has been turned in. This is a display-only field.
- **Turn In Amount** is the amount of the **Tender Type** that has been turned in. This is a display-only field.
- **Receipt Number** is the ID of the receipt given to the person who turn-in the funds. This is a display-only field.
- **Create Date/Time** contains when the turn-in event was created. This is a display-only field.

Maintaining Tender Controls

Tender controls exist to give you administrative control over your cash drawers (and all other tender sources).

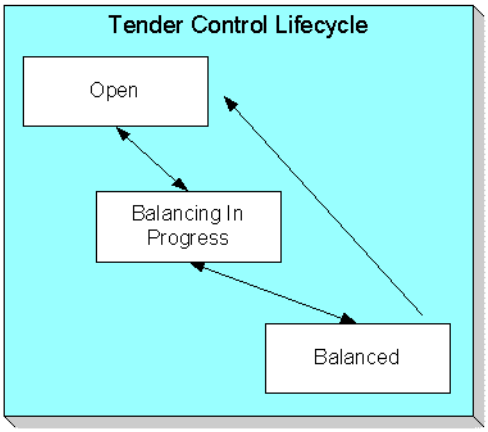
Fastpath: Refer to Managing Your Cash Drawers for more information.

Copyright © 2024, Oracle and/or its affiliates.

Note: The system creates most tender controls behind-the-scenes. Most tender controls are created by the system when it processes tenders from the remittance processors and lock boxes. You should only have to access the tender control pages if you record payments in cash drawers. For information about how the system creates tender controls, refer to Managing Payments Interfaced From External Sources. Also note that the automatic payment activation process also creates tender and deposit controls. Refer to Activating Automatic Payments for more information.

The Lifecycle Of A Tender Control

The following diagram shows the possible lifecycle of a tender control.



`Open` A tender control is initially created in the `Open` state. While in this state, you may add new tenders to it, change the tender amount on its tenders, and transfer tenders into and out of it.

`Balancing In Progress` You change a tender control's status to `Balancing In Progress` when you're ready to balance its contents. While in this state, you can change the tender amount on its tenders and transfer tenders out of it. If you need to add new tenders to it or transfer tenders into it, you must return it to the `Open` state.

`Balanced` You change a tender control's status to `Balanced` when the sum of its tenders is consistent with the ending balance in the drawer. While in this state, you cannot modify it or its tenders. If you need to make modifications, you must return it to the `Open` or `Balancing In Progress` state.

If the tender control is part of a `Balanced` deposit control, you may not change its status.

Note: Background processes and state transition. When payments are interfaced from external sources, the system automatically creates a tender control and links tenders to it (one for each payment interfaced). When all payments have been successfully loaded, the system changes the state of the respective tender control to `Balanced`.

Tender Control - Main

The Main page contains core tender control information. This page is used to balance the tender control.

Open this page using **Payments, Tender Control**.

Note: Searching For Tender Controls. When you use the **Creation Date** to search for tender controls, the system returns tender controls that are created on or before the date you specify. If the number of records returned exceeds the search limit (i.e. 300 records), only the records that do not exceed search limit are displayed. Therefore, you may need to specify an earlier creation date to find the record you are looking for. If you do not specify a creation date and search based on other criteria, the system uses the most recent date.

Fastpath: Refer to Managing Your Cash Drawers, Turn Ins and Balancing By Tender Type for more information.

Description of Page

Copyright © 2024, Oracle and/or its affiliates.

Tender Control contains a concatenation of the tender control's creation date, tender source, and status.

Tender Control ID is the system-assigned, unique identifier of the tender control.

Tender Source is the source of the tenders (e.g., cash drawer 22, lockbox 1 at Bank of America).

Deposit Control ID is the system-assigned, unique identifier of the deposit control that the tender control is part of. The deposit control's **Currency Code** is displayed below.

Fastpath: For more information about deposit controls and tender controls, refer to Managing Your Cash Drawers.

Turn on **All Users** if any user may insert tenders into this tender control. Turn this switch off and specify the appropriate **User ID** if only a single user may insert tenders into this tender control.

Create Date/Time is the date and time the tender control was created.

The **Starting Balance** of the tender control is defaulted from the tender source. This may be overridden while the **Tender Control Status** is `Open` or `Balancing in Progress`.

The **Tender Control Status** shows the status of the tender control. The **Balanced User ID** and **Balanced Date/Time** are populated when the status is changed to `Balanced`.

Fastpath: For more information, refer to The Lifecycle Of A Tender Control.

The **Balance** button is enabled when the **Tender Control Status** is `Balancing in Progress`. When this button is clicked, the system checks the following:

- All Turn-Ins have been `Approved`.
- The sum of all **Ending Bal** ances equals the sum of all **Expected Ending Bal** ances.

If the above conditions are true, the status of the tender control is changed to `Balanced` and all modifiable fields become protected.

The tenders grid contains a summary of the tenders linked to the tender control. One row is displayed for each tender type. Each row contains the number of tenders of this type and their total amount. In order to `Balance` the tender control, you must enter an appropriate **Ending Balance** for each **Tender Type**. Refer to Balancing By Tender Type for more information. The following information appears in the grid:

- **Tender Type** represents the type of tender (e.g., cash, credit card). This is a display-only field.
- **Nbr of Tenders** contains the total number of tenders of this type in the tender control. This is a display-only field and is calculated by the system by accumulating the tenders linked to the tender control.
- **Total Tenders** contains the total amount of tenders of this type in the tender control. This is a display-only field and is calculated by the system by accumulating the tenders linked to the tender control. Navigate to the **Tenders** page to see the individual tenders.
- **Total Turn Ins** contains the total amount of turn-ins of this type. This is a display-only field and is calculated by the system by accumulating the turn-ins linked to the tender control. Navigate to the **Turn Ins** pages to see the individual turn ins.
- **Starting Bal** contains the starting balance of this type of turn in. This field is only populated on the row defined as the Starting Balance tender type on the Installation Record. This is a display-only field and is equal to the tender control's **Starting Balance**.
- **Expected Ending Bal** is the **Starting Bal** anceplus **Total Tenders** minus **Total Turn Ins** for this type of tender.
- **Ending Bal** is the actual amount of money of this tender type. The cashier must enter this field in order to balance the tender control. This field is protected unless the **Status** of the tender control is `Balancing In Progress`. This amount must equal the Expected Ending Balance before the tender control can be marked as `Balanced`. Refer to How To Get An Unbalanced Tender Control In Balance (Fixing Over/Under) for more information.
- **Outstanding Over/Under** is the difference between **Expected Ending Bal** and **Ending Bal**. This is a display-only field.

The summary information beneath the tender type grid contains a summary of the tenders that are linked to the tender control.

- **Total Tender Amount** is the total amount of all tenders linked to the tender control.

Copyright © 2024, Oracle and/or its affiliates.

- **Ending Balance** is the actual amount of money in the tender control (as defined in the above grid).
- **Expected Ending Balance** is the expected amount of money in the tender control (as defined in the above grid).
- **Outstanding Over/Under** is Ending Balance minus Expected Ending Balance. If this amount is zero, the field is not displayed.

If the tender controls were created for a background process, such as the automatic payment extract, the following information is displayed:

- **Batch Code** is the batch code for which the tender control was created. This field only appears if it contains a value.
- **Batch Number** is the batch number for which the tender control was created. This field only appears if it contains a value.

Use **Comments** to describe anything unusual about the tender control (e.g., to explain why a large over/under amount was created).

Tender Control - Tenders

The Tenders page contains a row for every tender linked to the tender control. Open this page using **Payments, Tender Control, Tenders**.

Description of Page

Tender Control contains a concatenation of the tender control's creation date, tender source, and status.

Tender Control ID is the system-assigned, unique identifier of the tender control.

This grid at the top of the page contains a row for every tender linked to the tender control. When first displayed, the grid is ordered by the Create Date/Time column. No information about the tenders may be modified. To view or modify tender information, click the Go To button.

The grid at the bottom of the page contains a summary by tender type of all tenders linked to the tender control.

Tender Control - Turn Ins

Every time you turn in funds to the deposit control that is associated with your tender control, you create a turn-in event using this page. Open this page using **Payments, Tender Control** and navigate to the **Turn Ins** tab.

Fastpath: Refer to Managing Your Cash Drawers and Turn Ins for more information.

Description of Page

Tender Control contains a concatenation of the tender control's creation date, tender source, and status.

Tender Control ID is the system-assigned, unique identifier of the tender control.

The turn-ins grid contains one row for every turn-in event associated with the tender control. A turn-in event is created when moneys are physically transferred to the deposit control associated with the tender control. Turn-in events are approved on the Deposit Control - Turn Ins page. The following information is displayed in the grid:

- **Turn In Status** defines if the turn-in has been `Turned In` or `Approved` by the operator who is responsible for the Deposit Control. When a turn-in is first created, its status is `Turned In` and the following fields should be defined. Once the turn-in is `Approved` by the user responsible for the deposit control associated with the tender control, all of the following fields become display-only and the turn-in event cannot be deleted.
- **Tender Type** is the type of tender that has been turned in. This field may only be modified when the **Turn In Status** is `Turned In`.
- **Turn In Amount** is the amount of the **Tender Type** that has been turned in. This field may only be modified when the **Turn In Status** is `Turned In`.
- **Receipt Number** is the ID of the receipt given to the person who turn-in the funds. This field may only be modified when the **Turn In Status** is `Turned In`.
- **Create Date/Time** contains when the turn-in event was created. This is a display-only field.

Copyright © 2024, Oracle and/or its affiliates.

Tender Control - Exceptions

This page contains an entry for every exception (i.e., error) associated with the tender control's payments and payment events. Open this page using **Payments, Tender Control** and navigate to the **Exceptions** tab.

Fastpath: Refer to Exceptions for more information.

Description of Page

This page contains an entry for every exception (i.e., error) associated with the tender control's payments and payment events.

Tender Control - Characteristics

To update tender control characteristics, open **Payments, Tender Control** and navigate to the **Characteristics** tab.

Note: **Note:** If your tender control is balanced, you cannot update tender control characteristics.

Description of Page

The **Characteristics** collection contains information that describes miscellaneous information about the tender control.

The following fields display:

Characteristic Type Indicate the type of characteristic.

Sequence Controls the order in which characteristics of the same type are displayed.

Characteristic Value Indicate the value of the characteristic.

Interfacing Payments From External Sources

Most payments are NOT added by an operator using the Payment Event page. Rather, they are interfaced from an external source (e.g., a lock box or a remittance processor).

The base-package provides two interfaces to upload payments, each based on a different method of creating payment events.

Fastpath: Refer to Distributing A Payment Event for more information about how payment event distribution is handled in the system.

The topics in this section describe how these payment interfaces work.

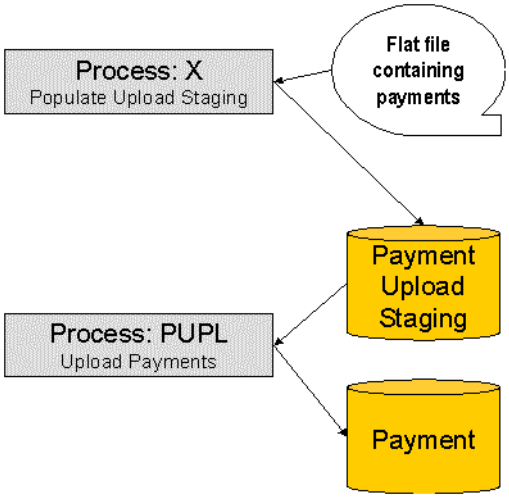
Interfacing Payments

The topics in this section describe how the payment interface using the system default method of creating payment events works.

Populating The Payment Upload Staging Records

The following diagram illustrates the processes involved in the uploading of payment into the system.

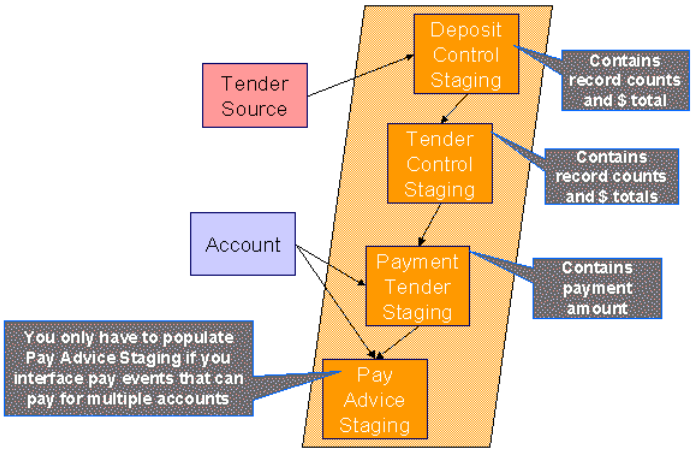
Copyright © 2024, Oracle and/or its affiliates.



The topics in this section describe each background process referenced above.

Process X - Populate Payment Upload Staging

Process X refers to the mechanism used by your organization to populate the various staging tables (shown in the orange section of the following ERD).



The topics in this section describe each of these tables.

Deposit Control Staging

You must create a deposit control staging record for each batch of payments to be uploaded into the system. The name of this table is CI_DEP_CTL_ST . The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
EXT_SOURCE_ID	30	Y	A/N	This must correspond with an external source ID on one of the defined tender sources. Refer to Setting Up Tender Sources for more information.
EXT_TRANSMIT_ID	30	Y	A/N	This is the unique identifier of the transmission from

Copyright © 2024, Oracle and/or its affiliates.

				the external source. This must be a unique value for each transmission from the source.
DEP_CTL_STG_ST_FLG	2	Y	A/N	This must be set to 20 (20 is the lookup value that corresponds with the `Pending` state)
DEP_CTL_ID	10	N	N	Leave this column blank. It will be assigned by the system when it creates a deposit control record.
TRANSMIT_DTTM	15	Y	DateTime	Date and time that the file was transmitted.
CURRENCY_CD	3	Y	A/N	This must be a valid currency code (this would be `USD` for United States dollars).
TOT_TNDR_CTL_AMT	32	Y	N	This column must equal the sum of the payment amounts on the tender control staging records associated with this deposit control staging.
TOT_TNDR_CTL_CNT	10	Y	N	This column must equal the number of tender control staging records associated with this deposit control staging.
LAST_UPDATE_INST	10	N	N	This field is populated during the upload. It is the process scheduler instance ID of the process performing the upload.

You must create one or more Tender Control Staging for this deposit control staging record.

Tender Control Staging

You must create at least one tender control staging record for each batch of payments to be uploaded into the system. The name of this table is CI_TNDR_CTL_ST . The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments

Copyright © 2024, Oracle and/or its affiliates.

EXT_SOURCE_ID	30	Y	A/N	This must correspond with the external source ID on the parent deposit control staging record.
EXT_TRANSMIT_ID	30	Y	A/N	This must correspond with the external transmission ID on the parent deposit control staging record.
EXT_BATCH_ID	30	Y	A/N	This is the unique identifier of the batch of payments in respect of the external transmission ID.
TNDR_CTL_STG_ST2_FLG		Y	A/N	This must be set to 20 (20 is the translate value that corresponds with the `Pending` state)
TNDR_CTL_ID	10	N	N	Leave this column blank. It will be assigned by the system when it creates a tender control record.
TOT_TNDR_AMT	13.2	Y	N	This column must equal the sum of the payment amounts on the payment tender staging records associated with this tender control staging.
TOT_TNDR_CNT	10	Y	N	This column must equal the number of payment tender staging records associated with this tender control staging.

You must create one or more Payment Tender Staging records for this tender control staging record.

Payment Tender Staging

You must create at least one payment tender staging record for each payment associated with the tender control staging record. The name of this table is CI_PAY_TNDR_ST . The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
EXT_SOURCE_ID	30	Y	A/N	This must correspond with the external

Copyright © 2024, Oracle and/or its affiliates.

				source ID on the parent deposit control staging record.
EXT_TRANSMIT_ID	30	Y	A/N	This must correspond with the external transmission ID on the parent tender control staging record.
EXT_BATCH_ID	30	Y	A/N	This must correspond with the external batch ID on the parent tender control staging record.
EXT_REFERENCE_ID	30	Y	A/N	This is the unique identifier of the payment in respect of the external batch ID.
PAY_TND_STG_ST_FLG	2	Y	A/N	This must be set to 10 (10 is the translate value that corresponds with the `Pending` state)
PAY_TENDER_ID	12	N	N	Leave this column blank. It will be assigned by the system when it creates a tender record.
TENDER_AMT	13.2	Y	N	The amount tendered (i.e., the payment amount).
ACCOUNTING_DT	10	Y	Date	This is the date that should be used for accounting purposes. This should correspond with an open accounting period.
TENDER_TYPE_CD	4	Y	A/N	This must correspond with the prime key of one of your tender types. Refer to Setting Up Tender Types for more information.
CUST_ID	15	Y	A/N	This is the account ID or old account number of the customer tendering the payment. If the system cannot find an account ID or old

Copyright © 2024, Oracle and/or its affiliates.

				account number that matches this value, the account ID of the tender source's suspense contract will be used on the corresponding tender and payment.
MICR_ID	30	N	A/N	This is the MICR ID associated with the payment.
NAME1	40	N	A/N	This is the customer name on the payment.
CHECK_NBR	10	N	A/N	This is the check number on the payment.

Payment Advice Staging

You need only populate rows on this table if any of the following conditions apply:

- If you need to distribute a payment tender to an account other than that defined with the CUST_ID on the payment tender staging record, you must create a payment staging record. You may distribute a tender to multiple accounts by creating multiple payment staging records. Note, if you want to distribute the payment tender to the same account, you do NOT need a payment staging record.
- If you want to restrict a payment to a specific contract, you must insert a row on this table to indicate the specific the contract in question. You do this by populating MATCH_TYPE_CD with a value that indicates that you are paying for a specific contract and MATCH_VALUE with the unique ID of the contract.
- If you practice open-item accounting, you must insert a row on this table for each to indicate the open-item to which the payment should be matched. Note, because open-item customer typically match payments to bills, you would populate MATCH_TYPE_CD with a value to indicate that you are matching by bill ID and MATCH_VALUE with the unique ID of the bill.

The name of this table is CI_PAY_ST . The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
EXT_SOURCE_ID	30	Y	A/N	This must correspond with the external source ID on the parent payment tender staging record.
EXT_TRANSMIT_ID	30	Y	A/N	This must correspond with the external transmission ID on the parent payment tender staging record.
EXT_BATCH_ID	30	Y	A/N	This must correspond with the external batch ID on the parent payment tender staging record.
EXT_REFERENCE_ID	30	Y	A/N	This must correspond with the external

Copyright © 2024, Oracle and/or its affiliates.

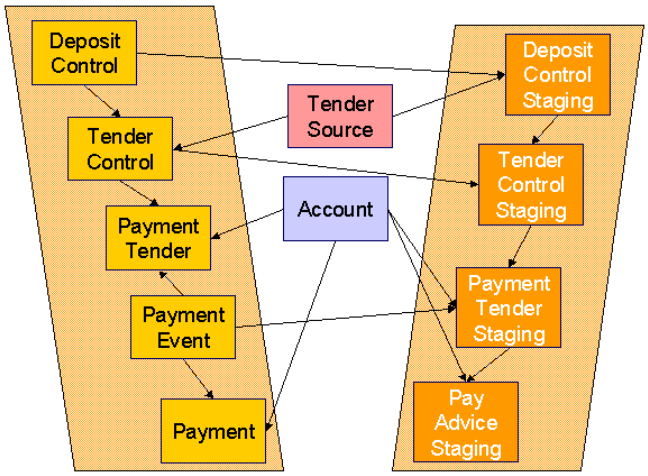
				reference ID on the parent payment tender staging record.
CUST_ID	15	Y	A/N	This is the account ID or old account number of the customer to which the payment should be distributed. If the system cannot find an account ID or old account number that matches this value, the account ID of the payor is used on the corresponding payment. If the payor's account ID is invalid, the tender source's suspense contract is used.
PAY_AMT	13.2	Y	N	The amount tendered (i.e., the payment amount).
MATCH_TYPE_CD	8	N	A/N	See the description of the MATCH_VALUE field below. Refer to Payments And Match Events for more information about the significance of this field.
MATCH_VALUE	30	N	A/N	MATCH_VALUE and MATCH_TYPE_CD are used in conjunction to indicate that the distribution of the payment should be restricted in some way (i.e., the standard payment distribution algorithm should not be used). MATCH_TYPE_CD indicates how the payment should be distributed (e.g., only distribute to a specific contract), MATCH_VALUE contains the ID of the

Copyright © 2024, Oracle and/or its affiliates.

				restriction (e.g., the contract ID). If MATCH_TYPE_CD is specified, it must reference a valid Match Type.

Process PUPL - Upload Payments

The batch process identified by the batch process ID `PUPL` refers to the background process that loads the contents of the various payment staging records into the various payment event tables. The tables that are populated by this process are shown in the left orange section of the following ERD (the right orange section are the staging tables populated by the process described above)



The topics in this section describe how these tables are populated.

Phase 1 - Create Deposit Control

The following points describe, at a high level, the first phase of the payment upload process:

- PUPL checks that the record counts and money totals of tender control stagings add up to the expected amount on deposit control staging. If not,
-
  - PUPL sets the status of the deposit control staging to be `Error`. None of the tender controls within the deposit control will be processed until everything adds up. You can fix these on the Deposit Control Staging page.
  - When PUPL runs next, it will recheck the totals on deposit control stagings that are in `Error` or `Pending`
- If the record and dollar amounts are clean,
-
  - PUPL creates the corresponding deposit control
  - PUPL sets the status of the deposit control staging to be `In Progress`

Phase 2 - Create Tender Control

The following points describe, at a high level, the second phase of the payment upload process:

- PUPL checks that record counts and money totals of payment tender staging(s) adds up to expected amount on tender control staging. If not,
-
  - PUPL sets the status of the tender control staging to be `Error`. None of the tender controls within the deposit control will be processed until everything adds up for ALL tender controls. You can fix these on the Tender Control Staging page.

Copyright © 2024, Oracle and/or its affiliates.

> **Note:** **Note** that the Deposit Control Staging record is **not** updated - its status is unchanged. Neither is the Pay Tender Staging record updated - its status also remains unchanged. Only the Tender Control Staging record is updated to be in `Error`.

- When PUPL runs next, it will recheck the totals of tender control stagings that are in `Error` or `Pending`.
- If the record and dollar amounts are clean,
- • PUPL creates the corresponding tender control.
  - PUPL sets the status of the tender control staging to be `In Progress`.

Phase 3 - Create Payment Events, Tenders, Payments and Payment Segments

At this point, all deposit control stagings and tender control stagings are in the state of `In Progress`. Next, PUPL starts the upload of payment tender staging and payment advice staging. The following points describe, at a high level, this phase of the payment upload process:

- If the payment tender staging record has a future accounting date, the processing for the record is skipped. This prevents uploaded payments from being created and subsequently frozen until their accounting date is reached. (Some external sources may provide advance notification of payments to be made in the future.) A skipped staging record remains in the `Pending` state until its accounting date is reached.
- PUPL checks money totals of payment advices (if any) adds up to expected amount on payment tender staging.
- • If not, PUPL sets the payment tender staging's status to `Error`.
  - Any errors are written to the Payment Upload Exception table. You can fix these errors on the Payment Upload Staging page and change the record's status back to `Pending`.
  - When PUPL runs next, it will recheck the totals of the payment tender staging
- If payment tender staging record is clean:
- • PUPL creates a corresponding payment event, tender, and payment.
  - If the account on payment tender staging is wrong, the account on the corresponding tender will be the tender source's suspense contract's account. Refer to Setting Up Tender Sources for more information. Refer to How To Transfer A Payment From One Account To Another for how to transfer to payment to the correct account.
  - If the account on payment advice is wrong, the account on the corresponding payment will be the account on the payment tender.
  - PUPL distributes the payment(s) amongst the account's contracts, and payment segments are created. Note, the payment could be in error if there are no contract's for the account (as well as other reasons). Payments in error are written to the Payment Exceptions table.
  - PUPL changes the payment tender staging's status to `Complete`.
- If all payment tender stagings are `Complete`:
- PUPL changes the tender control staging's status to `Complete`.
  - PUPL changes the deposit control staging's status to `Complete`.
  - If there are payment tender staging that are not `Complete`
  - • The status of the tender control staging will still be `In progress`.
    - The status of the deposit control staging will still be `In progress`.
  - PUPL will attempt to upload the offending payment tender staging records when it next runs.

PYUP-PRG - Purge Payment Upload Objects

`Completed` payment upload staging objects should be periodically purged from the system by executing the PYUP-PRG background process. This background process allows you to purge all `Completed` payment upload staging objects older than a given number of days.

We want to stress that there is no system constraint as to the number of `Completed` payment upload objects that may exist. You can retain these objects for as long as you desire. However we recommend that you periodically purge `Completed` payment upload objects as they exist only to satisfy auditing and reporting needs.

Maintaining Deposit Control Staging

The Deposit Control Staging page has three purposes:

Copyright © 2024, Oracle and/or its affiliates.

- You can view historical deposit and tender control staging records associated with uploaded payments.
- You can correct deposit and tender control records that are in error.
- You can add deposit and tender control records to be uploaded by the payment upload background process.

The topics in this section describe this page.

Deposit Control Staging - Main

This page shows the details of a deposit control staging record.

Fastpath: Refer to Populating The Payment Upload Staging Records for more information about this record.

Open this page using **Payments, Deposit Control Staging**.

Description of Page

External Source ID corresponds with an external source ID on one of the your tender sources. This should be the unique ID of the source of the interfaced payments. Refer to Setting Up Tender Sources for more information.

External Transmit ID is the unique identifier of the transmission of payments from the external source. This must be a unique value for each transmission from the source.

Status shows the state of the deposit control staging records. Potential values are: `Incomplete`, `Pending`, `In Progress`, `Partial Load`, `Complete`, `Error`.

Deposit Control ID is the system-assigned, unique identifier of the related deposit control. This value is populated after the system creates a deposit control for the upload staging record.

Transmission Date/ Time are when the information was interfaced into the system.

Total Tender Controls must equal the number of tender control staging records associated with this deposit control staging.

Total Tender Control Amount must equal the sum of the payment amounts on the tender control staging records associated with this deposit control staging. The **Currency Code** related to the amount is adjacent.

Deposit Control Staging - Tender Control Staging

This page shows the details of a tender control staging record.

Fastpath: Refer to Populating The Payment Upload Staging Records for more information about this record.

Open this page using **Payments, Deposit Control Staging, Tender Control Staging**.

Description of Page

External Source ID is the external source ID on the parent deposit control staging record.

External Transmit ID is the external transmission ID on the parent deposit control staging record.

The grid that follows contains a row for every tender control staging record linked to the deposit control staging record. The following information is displayed.

External Batch ID This is the unique identifier of the batch of payments in respect of the external transmission ID.

Status This is the state of the tender control staging records. Potential values are: `Pending`, `In Progress`, `Complete`, `Error`.

Tender Control ID This is the system-assigned, unique identifier of the related tender control. This value is populated after the system creates a tender control for the upload staging record.

Total Tenders Amount This is the sum of the payment amounts on the payment staging records associated with this tender control staging.

Total Number Of Tenders This is the number payment tender staging records associated with this tender control staging.

Copyright © 2024, Oracle and/or its affiliates.

Payment Upload Staging

The Payment Upload Staging page has three purposes:

- You can view historical payment tender and payment advice staging records associated with uploaded payments.
- You can correct payment tender records that are in error.
- You can add new payment tender and payment advice staging records to be uploaded by the payment upload process.

The topics in this section describe this page.

Payment Upload Staging - Tender Detail

This page shows the details of a payment tender staging record.

Fastpath: Refer to Populating The Payment Upload Staging Records for more information about this record.

Open this page using **Payments, Payment Upload Staging, Tender Detail**.

Description of Page

External Source ID this is the external source ID on the parent tender control staging record.

External Transmission ID is the external transmission ID on the parent tender control staging record.

External Batch ID is the external batch ID on the parent tender control staging record.

Ext. Reference ID is the external source's unique identifier of the payment tender.

Customer ID is the account ID or old account number of the customer tendering the payment. If the system cannot find an account ID or old account number that matches this value, the account ID of the tender source's suspense contract will be used on the corresponding tender and payment.

The **Tender Amount** is the amount tendered (i.e., the payment amount).

Tender Type defines the type of tender. Refer to Setting Up Tender Types for more

information.

Note: Authorizing Tenders. The **Tender Type** dropdown list does not include tender types that require authorization (i.e. credit card payments).

Fastpath: For information about tender types requiring authorization, refer to Credit Card Payments.

MICR ID is the MICR ID associated with the tender.

Check Number is the check number on the payment.

Name is the customer's name (as it appeared on the uploaded tender).

Accounting Date is the date that should be used for accounting purposes.

Pay Tender Staging Status shows the state of the tender control staging records. Potential values are: `Pending`, `Complete`, `Error`.

Payment Event ID is the system-assigned, unique identifier of the related payment event. This value is populated after the system creates a payment event for the upload staging record.

Payment Upload Staging - Payment Advice

If a tender was distributed to customers other than that defined on the Tender Detail page, the customers that the tender was distributed to are defined on this page. This page will not contain information if the tender is distributed to the tender detail's customer.

Fastpath: Refer to Populating The Payment Upload Staging Records for more information about this record.

Open this page using **Payments, Payment Upload Staging, Payment Advice**.

Copyright © 2024, Oracle and/or its affiliates.

Description of Page

External Source ID this is the external source ID on the parent tender control staging record.

External Transmission ID is the external transmission ID on the parent tender control staging record.

External Batch ID is the external batch ID on the parent tender control staging record.

External Reference ID is the external reference ID of the payment tender.

The grid that follows is only populated if the tender is distributed to customer(s) other than the tendering customer. The following information is displayed.

Customer ID This is the account ID or the old account number of the customer to which the payment should be distributed.

Customer Info If the **Customer ID** is an account ID that exists in the system, the name of the primary person and the customer class of the account are displayed here.

Payment Amount This is the amount of the tender to be distributed to the customer.

Match Type and **Match Value** These fields are only used if the distribution of the payment should be restricted in some way (i.e., the standard payment distribution algorithm should not be used). **Match Type** indicates how the payment should be distributed (e.g., only distribute to a specific contract), **Match Value** indicates the ID of the restriction (e.g., the contract ID). One of the valid values of **Match Type** is `Contract ID`.

Payment Upload Staging - Payment Characteristics

The **Payment Characteristics** tab allows you to define a characteristic for each payment advice. The **Left Arrow** (⇦) and **Right Arrow** (⇨) buttons allow you to navigate to the required payment advice. You can then define characteristics for the payment advice. This tab contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Effective Date	Used to specify the date from when the characteristic is effective for the payment advice.	Yes (Conditional) **Note:** This field is required when you are defining a characteristic for the payment advice.
Characteristic Type	Used to indicate the characteristic type. **Note:** The list includes only those characteristic types where the characteristic entity is set to Payment.	Yes (Conditional) **Note:** This field is required when you are defining a characteristic for the payment advice.
Sequence	Used to indicate the sequence number of the characteristic. This field enables you to define same characteristic with same value for a payment advice provided the sequence number is different.	Yes (Conditional) **Note:** This field is required when you are defining a characteristic for the payment advice.
Characteristic Value	Used to specify the value for the characteristic type.	Yes (Conditional)

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: On specifying the value for a predefined characteristic type, the description of the characteristic value appears corresponding to the **Characteristic Value** field. If you select a predefined characteristic type, the **Search** (🔍) icon appears corresponding to the **Characteristic Value** field. On clicking the **Search** icon, you can search for a predefined characteristic value.	**Note:** This field is required when you are defining a characteristic for the payment advice.

If you want to define more than one characteristic for the payment advice, click the **Add** (+) icon and enter the details. However, if you want to remove a characteristic from the payment advice, click the **Delete** (🗑) icon corresponding to the characteristic.

Note: You cannot remove a characteristic from a payment advice if the Pay Tender Staging Status is set to **Complete**.

Payment Upload Exception

If errors are detected during the payment upload process, a record is written to the payment upload exception table with a message indicating the nature of the severe error.

To view the messages associated with the exception records, schedule the TD-PYUPL background process. This process generates a To Do entry for every record in the payment upload exception table.

You can fix this error using the Payment Upload Staging page and change the status of the record from `Error` to `Pending`. When the payment upload process next runs, it attempts to upload this record again.

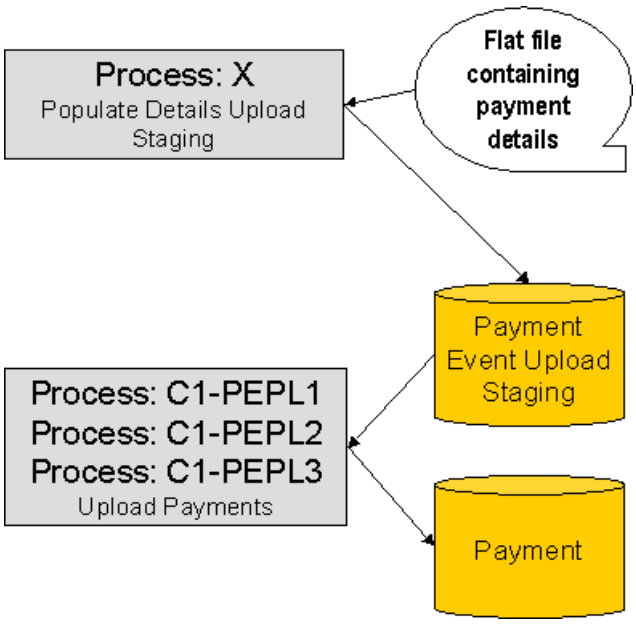
Interfacing Payments Using Distribution Rules

The topics in this section describe how the payment interface using distribution rules works.

Populating The Payment Event Upload Staging Records

The following diagram illustrates the processes involved in the uploading of payment event distribution details into the system.

Copyright © 2024, Oracle and/or its affiliates.



The topics in this section describe each background process referenced above.

Process X - Populate Payment Event Upload Staging

Process X refers to the mechanism used by your organization to populate the payment event upload staging table.

Payment Event Upload Staging Table

You must create a payment event upload staging record for each payment distribution detail to be uploaded into the system. The name of this table is CI_PEVT_DTL_ST . The following table describes each column on this table.

Column Name	Length	Req'd	Data Type	Comments
EXT_SOURCE_ID	30	Y	A/N	This must correspond with an external source ID on one of the defined tender sources. Refer to Setting Up Tender Sources for more information.
EXT_TRANSMIT_ID	30	Y	A/N	This is the unique identifier of the transmission from the external source. This must be a unique value for each transmission from the source.
PEVT_DTL_SEQ	12	Y	N	Unique identifier of the detail record within the transmission. The `C1-PEPL1` process uses this field to organize the parallel threads.
PEVT_STG_ST_FLG	4	Y	A/N	This must be set to 10 (10 is the lookup value that corresponds with the `Incomplete` state)
DST_RULE_CD	12	Y	A/N	This must be a valid distribution rule. The distribution rule contains the set of algorithms designed to process the staging detail.
DST_RULE_VALUE	254	Y	A/N	This must be a valid value for the characteristic type defined on the distribution rule.

Copyright © 2024, Oracle and/or its affiliates.

CURRENCY_CD	3	Y	A/N	This must be a valid currency code (this would be `USD` for United States dollars).
TENDER_AMT	13.2	Y	N	The amount tendered (i.e., the payment amount).
ACCOUNTING_DT	10	Y	Date	This is the payment date that should be used for accounting purposes. This should correspond with an open accounting period.
TENDER_TYPE_CD	4	Y	A/N	This must be a valid tender type. Refer to Setting Up Tender Types for more information.
CHECK_NBR	10	N	A/N	This is the check number on the payment.
MICR_ID	30	N	A/N	This is the MICR ID associated with the payment tender.
CUST_ID	15	N	A/N	This field may be used to record person information.
NAME1	30	N	A/N	This field may be used to capture additional payment tender information.
EXT_REFERENCE_ID	30	N	A/N	This field may be used to capture external information associated with the payment tender.
MATCH_TYPE_CD	8	N	A/N	See the description of the MATCH_VALUE field below. Refer to Payments And Match Events for more information about the significance of this field.
MATCH_VALUE	30	N	A/N	MATCH_VALUE and MATCH_TYPE_CD are used in conjunction to indicate that the distribution of the payment should be restricted in some way (i.e., the standard payment distribution algorithm should not be used). MATCH_TYPE_CD indicates how the payment should be distributed (e.g., only distribute to a specific contract), MATCH_VALUE contains the ID of the restriction (e.g., the contract ID). If MATCH_TYPE_CD is specified, it must reference a valid Match Type.
TNDR_CTL_ID	10	N	A/N	Leave this column blank. It will be assigned by the system when it creates a tender control record.
ACCT_ID	10	N	A/N	This is the tender account. If left blank, the `C1-PEPL1` process will populate this field by calling the `Determine Tender Account` algorithm defined on the distribution rule.

Copyright © 2024, Oracle and/or its affiliates.

				Note that this Account ID is not necessarily unique as multiple staging details can reference the same tender account.
PAY_EVENT_ID	12	N	A/N	Leave this column blank. It will be assigned by the system when it creates a payment event record.
PEVT_PROCESS_ID	10	N	A/N	If left blank, the `C1-PEPL1` process will set this field equal to the Tender Account ID. This field is used for grouping staging records **and** for organizing parallel threads (in the `C1-PEPL2` process). Therefore, it is **strongly encouraged** that it bears a relationship to the tender account ID.
APAY_SRC_CD	12	N	A/N	This must be a valid auto-pay source code.
EXT_ACCT_ID	50	N	A/N	This is the customer's account number at the financial institution
EXPIRE_DT	10	N	Date	This field is only needed if the Tender Type indicates that an expiration date is necessary (e.g., for a credit card payment)
ENTITY_NAME	64	N	A/N	This is the customer's name in the financial institution's system

Data Setup Examples of Payment Distribution Details

Typically, each staging record will represent a single payment event - with a corresponding payment tender and payment. However, it is possible to specify complex relationships within a set of staging records. For example, it will be straightforward to define a set of staging records that represent a single tender but multiple payments (to model the single payment tender of a welfare agency which covers payments for multiple accounts). Similarly, it is equally possible to define a set of staging records that represent a single payment but multiple tenders (although not a common requirement).

Each staging record can represent:

- Zero or one new payment events. Typically, each detail staging record will represent a single payment event. However, it is possible to define multiple records for a single payment event. All details for a single payment event are identified by a common value on the staging record: **Pay Event Process ID**.

Note: Pay Event Process ID. This field is used for grouping staging records **and** for organizing parallel threads (in the `C1-PEPL2` process). Therefore, it is **strongly encouraged** that it bears a relationship to the tender account ID.

- A partial or one new payment tenders. Typically, each detail staging record will represent a single payment tender. However, it is possible to define multiple staging records for a single pay tender. All details for a single tender are identified by a common payment event ID as well as common tender information (tender type, tender account ID, check number, external reference, etc.).
- A partial or many new payments. Typically, each detail staging record will represent a set of one or many new payments. However, it is possible to define multiple staging records for a single payment. All details for a single payment are identified by a common payment event ID as well as common distribution rule and characteristic value information.

The sections below provide examples of a few of these complex payment event configurations. Note that these examples assume the same distribution rule is referenced in all staging records.

Copyright © 2024, Oracle and/or its affiliates.

Staging Entry Example 1: One Payment Event, Two Tenders, Two Payments

Seq.	Pay Event Process ID	Tender Account ID	Tender Ref.	Date	Rule Value	Amount	Tender Type	Check No
1	1234567890	1234567890	112A	1/1/2006	113-54-8978	30.00	Check	101
2	1234567890	1234567890	112B	1/1/2006	575-40-3030	40.00	Check	102

Staging Entry Example 2: One Payment Event, One Tender, Two Payments

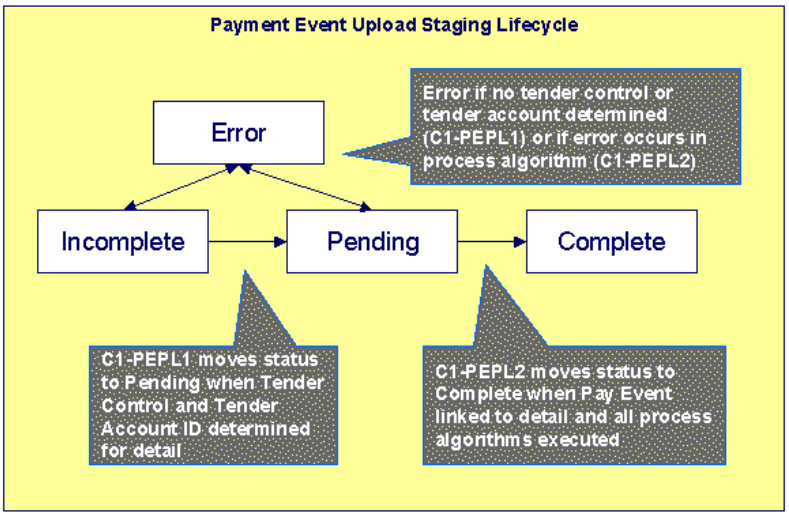
Seq.	Pay Event Process ID	Tender Account ID	Tender Ref.	Date	Rule Value	Amount	Tender Type	Check No
1	1234567890	1234567890	112A	1/1/2006	113-54-8978	30.00	Check	101
2	1234567890	1234567890	112A	1/1/2006	575-40-3030	40.00	Check	101

Staging Entry Example 3: One Payment Event, Two Tenders, One Payment

Seq.	Pay Event Process ID	Tender Account ID	Tender Ref.	Date	Rule Value	Amount	Tender Type	Check No
1	1234567890	1234567890	112A	1/1/2006	575-40-3030	30.00	Check	101
2	1234567890	7878787870	888Q	1/1/2006	575-40-3030	40.00	Check	9872

The Lifecycle of a Payment Event Upload Staging Record

The following diagram shows the possible lifecycle of a payment event upload staging record.



- `Incomplete`. A payment event staging record is initially created in `incomplete` state. The `C1-PEPL1` process sets it to `pending` once it links it to a tender control and determines its tender account.
- `Pending`. The `C1-PEPL2` process sets a `pending` record to `complete` once all processing logic is executed and a payment event is linked to it.
- `Complete`. When processing of the staging record is complete the record is in the `complete` state. This is a final state.
- `Error`. A payment event staging record may be set to `Error` from `Incomplete` or `Pending` states by the `C1-PEPL1` and `C1-PEPL2` processes respectively.

Copyright © 2024, Oracle and/or its affiliates.

Fastpath: Refer to ToDo Entries Instead of Exceptions for more information on how ToDo entries are used to capture processing errors.

To Do Entries Instead of Exception Records

Instead of creating an exception record for staging records in the `Error` state the `C1-PEPL1` and `C1-PEPL2` background processes create To Do entries and link them to the offending staging records.

Each process determines the To Do type to use for reporting errors by looking for the To Do Type defined with this process as its creation process.

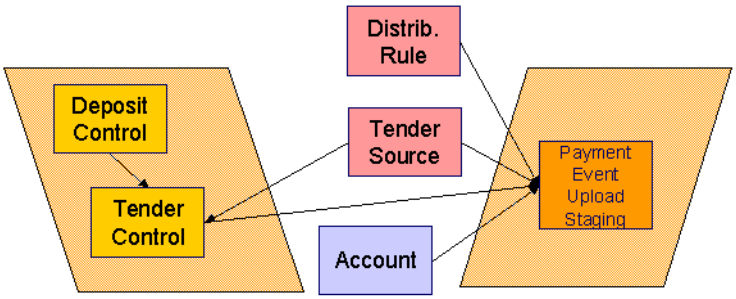
Process C1-PEPL1 - Upload Payments (Step 1)

The batch process `C1-PEPL1` refers to the first of three background processes that load the contents of the payment event upload staging records into the various payment tables.

The responsibility of this process is to transition the status of the staging records from `Incomplete` to `Pending`. The status of a staging record must remain `Incomplete` until:

- It is linked to a tender control. This process creates new deposit and tender control records, and then updates the payment event upload staging records with the corresponding **Tender Control ID**.
- The **Tender Account ID** field is populated. The `Determine Tender Account` algorithm defined on the distribution rule is called and the returned account ID is posted on the staging record.
- The **Pay Event Process ID** field is populated. If left blank this field is set equal to the **Tender Account ID**.

The following diagram and the sections below describe at a high level the processing phases of the `C1-PEPL1` background process.



Phase 1 - Create Tender Control

Note: This step cannot be bounded by thread range but must execute across the entire population of staging details. Therefore this step in the process is designed so that all parallel threads attempt to execute it at the same time but only one thread succeeds to avoid creating duplicate tender controls.

This step creates all tender controls required by the upload records as follows:

- For each distinct tender source transmissions represented within the `incomplete` set of upload staging details a deposit control, a deposit tender and a tender control are created in an `open` state. Note that a deposit tender record is only created if the tender amount is not zero.
- If an error occurs at this stage
- - A designated staging record for the transmission group (one is picked at random) is set to `Error` and a ToDo entry is created and linked to it to capture the error message that applies for the whole transmission group. Other records in the group remain `incomplete`.
  - The process stops.

Note: Group Error. This technique allows for an easier recovery from a setup error that may affect a large volume of records in a single transmission. Capturing the error only on a single designated record requires only this record to be set back to `Incomplete` once the setup issue is corrected. It is important to note that the transmission group is not processes if at least one of the records in the group is in `Error` status.

Copyright © 2024, Oracle and/or its affiliates.

- Once a transmission group of records is fully processed, a ToDo cleanup processing takes place to complete ToDo entries previously raised for its designated staging record.

Fastpath: Refer to ToDo Entries Instead of Exceptions for more information on how ToDo entries are used to capture processing errors.

Phase 2 - Determine Tender Account

After all tender controls have been created the second step attempts to transition all `incomplete` records to the `pending` state.

Each `incomplete` staging record is processed as follows:

- If not yet associated with a tender control ID, the process looks for an `Open` tender control that matches the batch code, batch number and external source ID and links it to the staging record. An error is raised if a matching tender control is not found.
- Execute the `Determine Tender Account` algorithm defined on the distribution rule and populate the record with the returned account ID. Note that the algorithm is called even when the tender account is populated to provide for a potential override of the initial value when necessary. An error is raised if a tender account may not be determined.
- If not already populated, the **Pay Event Process ID** is set equal to the Tender Account ID. The `C1-PEPL2` process uses this field to organize the parallel threads and to group multiple staging details into a single payment event.
- The staging record is moved to `pending` state.
- If any errors occur set the record to `Error` and create a To Do entry for the error message.
- If no error occurred, a ToDo cleanup processing takes place to complete ToDo entries previously raised for the staging record.

Fastpath: Refer to ToDo Entries Instead of Exceptions for more information on how ToDo entries are used to capture processing errors.

Note: **Note.** This step is designed to support execution in parallel threads based upon the sequence number portion of the staging table prime key.

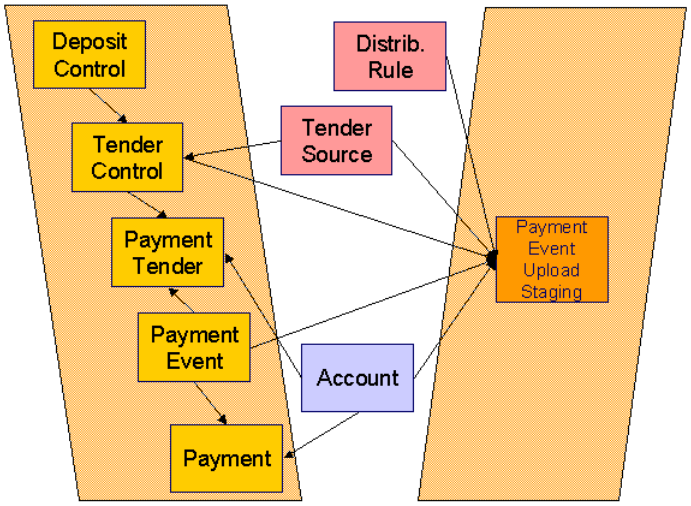
Process C1-PEPL2 - Upload Payments (Step 2)

The batch process `C1-PEPL2` refers to the second of three background processes that load the contents of the payment event upload staging records into the various payment tables.

The responsibility of the `C1-PEPL2` process is to create payment events, payment tenders and payments and transition the corresponding staging records from `Pending` to `Complete`.

The following diagram and the section below describe at a high level the processing phases of the `C1-PEPL2` background process.

Copyright © 2024, Oracle and/or its affiliates.



Each distinct group of `pending` staging records associated with the same external source ID, external transmit ID, accounting date, and pay event process ID is processed as follows:

- A payment event is created for the group and stamped on each of its records.
- A payment tender is created for each distinct set of staging records having the same tender account, tender type and other tender information fields, except for the tender amount.
- The `Create Payment` algorithm is then called for each distinct group of staging records having the same tender account, distribution rule and rule value providing it with the total amount for the group.
- The staging record is moved to `pending` state.
- If any error occurs a designated staging record for the payment event group (one is picked at random) is set to `Error` and a To Do entry is created and linked to it to capture the error message that applies for the whole group. Other records in the group remain `incomplete`.

Note: Group Error. This technique allows for an easier recovery from an error that affects all staging records for a single payment event. Capturing the error only on a single designated record requires only this record to be set back to `pending` once the issue is corrected. It is important to note that the whole set of records is not processes if at least one of the records in the group is in `Error` status.

- If no error occurred, a To Do cleanup processing takes place to complete To Do entries previously raised for the designated staging record.

Fastpath: Refer to ToDo Entries Instead of Exceptions for more information on how To Do entries are used to capture processing errors.

Note: Note. This process is designed to support execution in parallel threads based upon the payment event process ID field.

Process C1-PEPL3 - Upload Payments (Step 3)

The batch process `C1-PEPL3` refers to the last of three background processes that load the contents of the payment event upload staging records into the various payment tables.

The responsibility of the `C1-PEPL3` process is to update the status of the related deposit and tender controls from `open` to `balanced`.

Each distinct tender control for which all associated staging records are in `complete` status is processed as follows:

- The tender control is set to `balanced`.
- The deposit control is set to `balanced`.

Note: Note. This process is designed to support execution in parallel threads based upon the Tender Control ID field.

Copyright © 2024, Oracle and/or its affiliates.

Payment Event Upload Staging

The Payment Event Upload Staging page has three purposes:

- You can view historical payment event upload staging records associated with uploaded payments.
- You can correct payment event upload staging records that are in `error`.
- You can add new payment event upload staging records to be uploaded by the payment event upload processes.

The topics in this section describe this page.

Payment Event Upload Staging - Main

This page shows the details of a payment event upload staging record.

Fastpath: Refer to Populating The Payment Event Upload Staging Records for more information about this record.

Open this page using **Payments, Payment Event Upload Staging**.

Description of Page

External Source ID corresponds with an external source ID on one of the your tender sources. This should be the unique ID of the source of the interfaced payments. Refer to Setting Up Tender Sources for more information.

External Transmit ID is the unique identifier of the transmission of payments from the external source. This must be a unique value for each transmission from the source.

Sequence is the identifier of the record within the transmission. The `C1-PEPL1` process uses this field to organize the parallel threads.

Accounting Date is the payment date that should be used for accounting purposes.

Distribution Rule is the rule by which the payment detail is to be processed. A default distribution rule is displayed if you have set one.

Rule Value is a value associated with the payment and expected by the distribution rule.

Match Type and **Match Value** are only used if the distribution of the payment should be restricted in some way (i.e., the standard payment distribution algorithm should not be used). **Match Type** indicates how the payment should be distributed (e.g., only distribute to a specific contract), **Match Value** indicates the ID of the restriction (e.g., the contract ID).

Payor Account ID is the tender account. If not populated the `C1-PEPL1` process populates this field by calling the `Determine Tender Account` algorithm defined on the distribution rule.

Tender Amount is the amount tendered (i.e., the payment amount).

Currency Code is the currency of the tendered amount. This should be the same as currency defined on the tender source.

Tender Type defines the form or remittance (e.g., cash, check, etc.). Note that the **Tender Type** defaults from the installation record.

Note: Authorizing Tenders. The **Tender Type** dropdown list does not include tender types that require authorization (i.e. credit card payments).

Fastpath: For information about tender types requiring authorization, refer to Credit Card Payments.

Check Number is the check number on the payment.

MICR ID is the value of the magnetic ink character recognition (MICR) line on the payment.

External Reference ID may be used to record external information associated with the payment tender.

Customer ID may be used to record additional customer information.

Name may be used to record additional payment tender information.

Copyright © 2024, Oracle and/or its affiliates.

Tender Control ID is the tender control associated with the payment. This field is should typically be left for the `C1-PEPL1` process to populate.

If the **Tender Type** is associated with an automatic payment, the **Auto Pay** section displays. The system attempts to default automatic payment information from the account's auto-pay option if the tender type is the same as the tender type on the account's auto-pay source and if the auto pay option is effective on the payment date. If the system is unable to default information, you must specify the source of the funds and the customer's account number / credit card number at the financial institution.

- **Auto Pay Source Code** is the financial institution / credit card company that receives the automatic payment request.
- **External Account ID** is the customer's account number at the financial institution.
- **Expires On** is only needed if the **Tender Type** indicates that an expiration date is necessary (e.g., for a credit card payment).
- **Name** is the customer's name in the financial institution's system.

Payment Process Event ID is used to group multiple staging records into a single payment event. If not populated, the `C1-PEPL1` process sets this field equal to the **Payor Account ID**.

Pay Tender Staging Status shows the state of the staging record. Refer to The Lifecycle of a Payment Event Upload Staging for a state transition overview.

Payment Event ID is the system-assigned, unique identifier of the related payment event. The `C1-PEPL2` process populates this field when it creates a payment event for the upload staging record. You can use this field to navigate to the payment event page.

If a staging record is in `Error` state then the error message associated with the corresponding To Do entry is displayed.

Payment Creation, Cancellation, Transfer, and View Through Inbound Web Service

Oracle Revenue Management and Billing enables you to create, transfer, cancel, and view payments through a web service. You can send the payment creation, payment transfer, payment cancellation, or payment view request from the external system through an inbound web service. The following four business services are shipped with the product to support different payment operations:

Business Service	Description
C1-CreatePayment	Used to create payments for an account. While sending the payment creation request, you need to specify either of the following of the payor and payee accounts: • Account ID • Account Identifier Type, Account Identifier, and Division The system enables you to create a payment through the web service using either of the following match types: • Bill • Contract You can specify maximum four tender types along with the amount. You can specify maximum four characteristics for payment tender and maximum five characteristics for payment. On processing the payment creation request, the system calculates the sum of tender amount and sets it as the payment amount. It creates the payment event and payments. The payments are immediately distributed and frozen. It also creates a payment tender for each tender type.

Copyright © 2024, Oracle and/or its affiliates.

Business Service	Description
	If the number of bill segments in a bill exceeds the bill segment count specified in the **C1_PAYBYBTCH** feature configuration, the system distributes and freezes the payments in the deferred mode and not in real time (i.e. immediately). The system creates a batch job using the **C1-PAYBT** batch and accordingly sends the response to the external system. For more information, see C1-CreatePayment Business Service on page 415
C1-CancelPayment	Used to cancel a payment or all payments of a payment event. While sending the payment cancellation request, you can specify either of the following cancellation types: • **T** - Used to indicate that you want to cancel all payments of a payment event along with its payment tenders. • **P** - Used to indicate that you want to cancel a particular payment. On processing the payment cancellation request, the system cancels the payment and changes the status of the payment to **Canceled**. For more information, see C1-CancelPayment Business Service on page 431
C1-TransferPayment	Used to transfer a payment or all payments of a payment event. While sending the payment transfer request, you can specify either of the following transfer types: • **E** - Used to indicate that you want to transfer all frozen payments of a payment event. • **P** - Used to indicate that you want to transfer a particular payment. The system enables you to transfer a payment through the web service using either of the following match types: • Bill • Contract You need to specify a reason why you want to transfer a payment. On processing the payment transfer request, the system cancels the old payment and creates a new payment on the payment event in the **Frozen** status. For more information, see C1-TransferPayment Business Service on page 437
C1-ViewPayment	Used to view the payments of an account. While sending the payment view request, you need to specify either of the following: • Account ID • Account Identifier Type, Account Identifier, and Division You can specify various criteria (such as, tender type, check number, payment creation date range) along with the account details to filter the list of payments. You can specify the maximum number of records that can be retrieved through the **C1-ViewPaymentService** business service in the **Set Fetch Record Limit** option type of the **C1-RECLMT** feature configuration. Accordingly, the system lists the number of payments in the payment view response. For example, if the value in the **Set Fetch Record Limit** option type is set to 100 and the result set contains 125 payments, the system will list only 100 payments of the account from the result set in the descending order of the creation date. For more information, see C1-ViewPayment Business Service on page 443

The system enables you to track the payment creation, payment transfer, payment cancellation, and payment view responses sent to the external system using the **Inbound Web Service History** feature.

We have not shipped any inbound web service with the product. You can create one inbound web service which supports all four operations - payment creation, payment transfer, payment cancellation, and payment view. Alternatively, you can create four inbound web services - one for payment creation, second for payment transfer, third for payment cancellation, and fourth for payment view.

Copyright © 2024, Oracle and/or its affiliates.

You can create a SOAP or REST based inbound web service. If you want to send the request in the XML and/or JSON format, you need to ensure that you create a REST based inbound web service. However, if you want to send the request in the XML format, you can create a SOAP or REST based inbound web service.

Prerequisites

To create the payment instructions through an inbound web service, you need to do the following:

- Define a REST or SOAP based inbound web service to call the **Create Payment Service (C1-CreatePaymentService)** business service.
- Define the required payer and payee account in the system.
- Define the required tender source, tender control, and tender types in the system.
- Define the deposit control in the system.
- Define the required match types in the system.
- Define the required payment cancellation reasons in the system.
- Ensure that bills are in the **Completed** status and the contracts are in the **Active** status.

Related Topics

For more information on...	See...
How to setup the **C1-RECLMT** feature configuration	Setting the C1-RECLMT Feature Configuration

C1-CreatePayment Business Service

The **Create Payments (C1-CreatePaymentService)** business service is used for creating payments in the Oracle Revenue Management and Billing application based on the data provided from the external systems through an inbound web service. This business service will use the existing payments services **CILPEVTP** and **CILPPAYP** for payment creation and event purposes.

While creating a payment, you need to provide a match value (such as a bill or contract) based on which the system summarizes all of the tender amounts in the system to be used as the payment amount to create the payment.

Note: The create payment business service distributes and freezes the payment based on the match type, provided in the form of an input to the business service.

Depending on the number of tender types, one or more **Pay Tender Ids** are created in the system. The system won't allow more than four tenders and five payment tender characteristics as an input to the business service.

Note: You have to set **M** as the value for the **CI_PAY_EVENT** and **PAY_EVENT_MODE** payment events, as the payment related information can be easily viewed on the **Payment Event AWB** screen.

Once the payment is successfully created, this business service responds with the following fields:

- Payment ID
- Payment Event ID
- Payment Status (Frozen or Error)
- Response Message (Success / Failure)

Copyright © 2024, Oracle and/or its affiliates.

Payment Creation Request in XML Format

The following code snippet illustrates how the tags should be nested while receiving the payment creation request in the XML format:

```
<c1-createPayment>
    <payeeAccount></payeeAccount>
    <accountNumberPayee></accountNumberPayee>
    <accountNumberTypeCdPayee></accountNumberTypeCdPayee>
    <divisionCodePayee></divisionCodePayee>
    <matchType></matchType>
    <matchTypeValue></matchTypeValue>
    <statementId></statementId>
    <paymentDt></paymentDt>
    <sourceSystem></sourceSystem>
    <externalTransactionId></externalTransactionId>
    <externalSourceId></externalSourceId>
    <tenderList>
        <tenderType></tenderType>
        <tenderCurrency></tenderCurrency>
        <tenderAmount></tenderAmount>
        <checkNumber></checkNumber>
        <payorAccount></payorAccount>
        <accountNumber></accountNumber>
        <accountNumberTypeCd></accountNumberTypeCd>
        <divisionCode></divisionCode>
        <micrId></micrId>
        <externalReferenceId></externalReferenceId>
        <name></name>
        <paymentTenderCharacteristic>
            <characteristicType></characteristicType>
            <characteristicValue></characteristicValue>
        <paymentTenderCharacteristic>
   </tenderList>
</c1-createPayment>
```

Payment Creation Response in XML Format

The following code snippet illustrates how the tags are nested while sending the payment creation response in the XML format:

```
<C1-createPayment>
    <payeeAccount></payeeAccount>
    <matchType></matchType>
    <matchTypeValue></matchTypeValue>
    <statementId></statementId>
    <tenderList>
        <tenderType></tenderType>
        <tenderCurrency></tenderCurrency>
        <tenderAmount></tenderAmount>
        <checkNumber></checkNumber>
        <payorAccount></payorAccount>
        <micrId></micrId>
        <externalReferenceId></externalReferenceId>
        <name></name>
        <paymentTenderCharacteristic>
            <characteristicType></characteristicType>
            <characteristicValue></characteristicValue>
        <paymentTenderCharacteristic>
   </tenderList>
    <paymentCharacteristic>
        <characteristicType></characteristicType>
```

Copyright © 2024, Oracle and/or its affiliates.

```
        <characteristicValue></characteristicValue>
    <paymentCharacteristic>
    <paymentsDtls>
        <paymentEventId></paymentEventId>
        <paymentId></paymentId>
        <messageValue></messageValue>
        <paymentStatusDesc></paymentStatusDesc>
        <statusCd></statusCd>
        <paymentStatus></paymentStatus>
    </paymentsDtls>
</C1-createPayment>
```

Payment Creation Request in JSON Format

The following code snippet illustrates how the tags should be nested while receiving the payment creation request in the JSON format:

```
{
    -"C1-CreatePaymentService":
    {
        -"accountNumberPayee": -"",
        -"accountNumberTypeCdPayee": -"",
        -"divisionCodePayee": -"",
        -"matchType": -"",
        -"matchTypeValue": -"",
        -"statementId": -"",
        -"currencyCd": -"",
        -"paymentDt": -"",
        -"sourceSystem": -"",
        -"externalTransactionId": -"",
        -"externalSourceId": -"",
        -"tenderList":
        [
            {
                -"tenderType": -"",
                -"tenderCurrency": -"",
                -"tenderAmount": -,
                -"checkNumber": -"",
                -"payorAccount": -"",
                -"accountNumber": -"",
                -"accountNumberTypeCd": -"",
                -"divisionCode": -"",
                -"micrId": -"",
                -"externalReferenceId": -"",
                -"name": -"",
                -"paymentTenderCharacteristic":
                [
                    {
                        -"characteristicType": -"",
                        -"characteristicValue": -""
                 -}
                -],
             -"paymentTenderCharacteristic":
                [
                    {
                        -"characteristicType": -"",
                        -"characteristicValue": -""
                    -}
                -]
            -}
        -],
      -"tenderList":
```

Copyright © 2024, Oracle and/or its affiliates.

```
        [
            {
                -"tenderType": -"",
                -"tenderCurrency": -"",
                -"tenderAmount": -,
                -"checkNumber": -"",
                -"payorAccount": -"",
                -"accountNumber": -"",
                -"accountNumberTypeCd": -"",
                -"divisionCode": -"",
                -"micrId": -"",
                -"externalReferenceId": -"",
                -"name": -"",
                -"paymentTenderCharacteristic":
                [
                    {
                        -"characteristicType": -"",
                        -"characteristicValue": -""
                    -}
                -]
            -}
        -],
        -"paymentCharacteristic":
        [
            {
                -"characteristicType": -"",
                -"characteristicValue": -""
            -}
        -],
        -"paymentCharacteristic":
        [
           {
               -"characteristicType": -"",
            -"characteristicValue": -""
        -}
        -]
    -}
}
```

Payment Creation Response in JSON Format

The following code snippet illustrates how the tags are nested while sending the payment creation response in the JSON format:

```
{
    -"C1-CreatePaymentService":
    {
        -"accountNumberPayee": -"",
        -"accountNumberTypeCdPayee": -"",
        -"divisionCodePayee": -"",
        -"matchType": -"",
        -"matchTypeValue": -"",
        -"statementId": -"",
        -"paymentDt": -"",
        -"tenderList":
        [
            {
                -"tenderType": -"",
                -"tenderCurrency": -"",
                -"tenderAmount": -"",
                -"checkNumber": -"",
                -"payorAccount": -"",
```

Copyright © 2024, Oracle and/or its affiliates.

```
                -"micrId": -"",
                -"externalReferenceId": -"",
                -"paymentTenderCharacteristic":
                [
                    {
                        -"characteristicType": -"",
                        -"characteristicValue": -""
                    -},
                    {
                        -"characteristicType": -"",
                        -"characteristicValue": -""
                    -}
                -]
            -},
            {
                -"tenderType": -"",
                -"tenderCurrency": -"",
                -"tenderAmount": -"",
                -"checkNumber": -"",
                -"payorAccount": -"",
                -"micrId": -"",
                -"externalReferenceId": -"",
                -"paymentTenderCharacteristic":
                {
                    -"characteristicType": -"",
                    -"characteristicValue": -""
                -}
            -}
        -],
        -"paymentCharacteristic":
        [
            {
                -"characteristicType": -"",
                -"characteristicValue": -""
            -},
            {
                -"characteristicType": -"",
                -"characteristicValue": -""
            -}
        -],
        -"paymentsDtls":
        {
            -"paymentEventId": -"",
            -"paymentId": -"",
            -"messageValue": -"",
            -"paymentStatusDesc": -"",
            -"paymentStatus": -""
        -}
    -}
}
```

Payment Creation Request and Response - Tags in XML Format

Note: We recommend you to refer the Payment Creation Request in XML Format on page 416 andPayment Creation Response in XML Format on page 416 topics in parallel while understanding the below mentioned tags. This will help you to understand how the tags are nested in the XML format.

Before calling the **C1-CreatePaymentService** business service through an inbound web service, you need to ensure that the create payment request contains the following tags:

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Tag Description	Mandatory (Yes or No)
payeeAccount	Used to indicate the account for which you want to create a payment.	Yes (Conditional) **Note:** This data is required if **accountNumberPayee** and **accountNumberTypeCdPayee** is not specified.
accountNumberPayee	Used to specify the value for the account identifier type of the payee. **Note:** You must specify the account number which is already defined in the system.	Yes (Conditional) **Note:** This data is required if **accountNumberTypeCdPayee** is specified.
accountNumberTypeCdPayee	Used to specify the account identifier type of the payee.	Yes (Conditional) **Note:** This data is required while creating a payment.
divisionCodePayee	Used to indicate the division code of the payee. **Note:** You must indicate the division code which is already defined in the application.	Yes (Conditional) **Note:** This data is required while creating a payment.
matchType	Used to indicate the match type using which you want to match the payment. **Note:** You must indicate the match type which is already defined in the application.	Yes
matchTypeValue	Used to specify a particular entity (such as bill ID or contract ID) for the match type.	Yes

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Tag Description	Mandatory (Yes or No)
	Note: The match value must be a valid bill ID or contract ID. If you specify a match type where the entity type is set to **Account**, the match value should not be provided. The payee account ID is directly fetched from the dataset. If the payee account ID is not available in the dataset, it is derived using the payee account identifier type, identifier value, and division. The system then derives the unpaid bills of the payee account using the manual distribution algorithm attached to the respective match type. If you specify a match type where the entity type is set to **Statement**, the match value should not be provided. The statement ID is directly fetched from the dataset. The system then derives the unpaid bills of the statement using the manual distribution algorithm attached to the respective match type.	
statementId	Used to specify the statement ID.	Yes (Conditional) **Note:** This data is required while distributing the payment amount against the unpaid bills of the statement.
paymentDt	Used to specify the date when the payment was made. **Note:** If the payment date is not provided, system date is considered.	No
sourceSystem	Used to indicate the external system from where the create payment request is received.	No
externalTransactionId	Used to indicate the external transaction ID of the payment.	No
externalSourceId	Used to specify the external source system ID.	No
tenderList	Used to specify the tender details used for creating a payment.	No
tenderType	Used to indicate the type of tender in which a payment is created. **Note:** You must at least select one tender type.	Yes
tenderCurrency	Used to indicate the currency in which a payment is created.	No
tenderAmount	Used to specify the tender amount. **Note:** The tender amount must be a positive amount.	Yes
checkNumber	Used to specify the check number, if payment is made through check.	No
payorAccount	Used to indicate the payor account for which you want to create the payment.	No

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Tag Description	Mandatory (Yes or No)
accountNumber	Used to indicate the bank account number through which the payment is created.	No
accountNumberTypeCd	Used to specify the account number type code of the payor.	No
divisionCode	Used to indicate the division code. **Note:** You must specify the division code which is already defined in the system.	Yes (Conditional) **Note:** This data is required while creating the payment.
micrId	Used to specify the MICR code on the payment.	No
externalReferenceId	Used to specify the external reference number of the payment.	No
name	Used to indicate the person for whom the payment is created.	No
paymentTenderCharacteristic	Used to specify the characteristic details of the payment and tender.	No
characteristicType	Used to indicate the characteristic type.	Yes (Conditional) **Note:** This field is required when you are defining a characteristic for the payment tender.
characteristicValue	Used to specify the value for the characteristic type.	Yes (Conditional) **Note:** This field is required when you are defining a characteristic for the payment tender.

The following table provides information about the response tags mentioned in the payment creation business service:

Tag Name	Tag Description
paymentsDtls	Displays the payment details.
paymentEventId	Displays the payment event Id.
paymentId	Displays the payment Id.
messageValue	Indicates an error that occurred while creating a payment request.
paymentStatusDesc	Indicates the reason why the create payment process is rejected.
statusCd	Indicates whether the payment creation request is successfully processed or not. The valid values are: • Success • Fail
paymentStatus	Indicates the payment status. The valid values are: • Frozen • Error

Copyright © 2024, Oracle and/or its affiliates.

Related Topics

For more information on...	See...
Sample Payment Creation Request and Response in XML Format	Sample Payment Creation Request and Response in XML Format on page 426

Payment Creation Request and Response - Attribute/Value Pairs in JSON Format

Note: We recommend you to refer thePayment Creation Request in JSON Format on page 417 andPayment Creation Response in JSON Format on page 418 topics in parallel while understanding the below mentioned attributes. This will help you to understand how the attributes are nested in the XML format.

Before calling the **C1-CreatePaymentService** business service through an inbound web service, you need to ensure that the create payment request contains the following attributes:

Attribute Name	Attribute Description	Mandatory (Yes or No)
payeeAccount	Used to indicate the account for which you want to create a payment.	Yes (Conditional) **Note:** This data is required if **accountNumberPayee** and **accountNumberTypeCdPayee** is not specified.
accountNumberPayee	Used to specify the value for the account identifier type of the payee. **Note:** You must specify the account number which is already defined in the system.	Yes (Conditional) **Note:** This data is required if **accountNumberTypeCdPayee** is specified.
accountNumberTypeCdPayee	Used to specify the account identifier type of the payee.	Yes (Conditional) **Note:** This data is required while creating a payment.
divisionCodePayee	Used to indicate the division code of the payee. **Note:** You must indicate the division code which is already defined in the application.	Yes (Conditional) **Note:** This data is required while creating a payment.
matchType	Used to indicate the match type using which you want to match the payment. **Note:** You must indicate the match type which is already defined in the application.	Yes
matchTypeValue	Used to specify a particular entity (such as bill ID or contract ID) for the match type.	Yes

Copyright © 2024, Oracle and/or its affiliates.

Attribute Name	Attribute Description	Mandatory (Yes or No)
	Note: The match value must be a valid bill ID or contract ID. If you specify a match type where the entity type is set to **Account**, the match value should not be provided. The payee account ID is directly fetched from the dataset. If the payee account ID is not available in the dataset, it is derived using the payee account identifier type, identifier value, and division. The system then derives the unpaid bills of the payee account using the manual distribution algorithm attached to the respective match type. If you specify a match type where the entity type is set to **Statement**, the match value should not be provided. The statement ID is directly fetched from the dataset. The system then derives the unpaid bills of the statement using the manual distribution algorithm attached to the respective match type.	
statementId	Used to specify the statement ID.	Yes (Conditional) **Note:** This data is required while distributing the payment amount against the unpaid bills of the statement.
paymentDt	Used to specify the date when the payment was made. **Note:** If the payment date is not provided, system date is considered.	No
sourceSystem	Used to indicate the external system from where the create payment request is received.	No
externalTransactionId	Used to indicate the external transaction ID of the payment.	No
externalSourceId	Used to specify the external source system ID.	No
tenderList	Used to specify the tender details used for creating a payment.	No
tenderType	Used to indicate the type of tender in which a payment is created. **Note:** You must at least select one tender type.	Yes
tenderCurrency	Used to indicate the currency in which a payment is created.	No
tenderAmount	Used to specify the tender amount. **Note:** The tender amount must be a positive amount.	Yes
checkNumber	Used to specify the check number, if payment is made through check.	No
payorAccount	Used to indicate the payor account for which you want to create the payment.	No

Copyright © 2024, Oracle and/or its affiliates.

Attribute Name	Attribute Description	Mandatory (Yes or No)
accountNumber	Used to indicate the bank account number through which the payment is created.	No
accountNumberTypeCd	Used to specify the account number type code of the payor.	No
divisionCode	Used to indicate the division code. **Note:** You must specify the division code which is already defined in the system.	Yes (Conditional) **Note:** This data is required while creating the payment.
micrId	Used to specify the MICR code on the payment.	No
externalReferenceId	Used to specify the external reference number of the payment.	No
name	Used to indicate the person for whom the payment is created.	No
paymentTenderCharacteristics	Used to specify the characteristic details of the payment and tender.	No
characteristicType	Used to indicate the characteristic type.	Yes (Conditional) **Note:** This field is required when you are defining a characteristic for the payment tender.
characteristicValue	Used to specify the value for the characteristic type.	Yes (Conditional) **Note:** This field is required when you are defining a characteristic for the payment tender.

The following table provides information about the response attributes mentioned in the payment creation business service:

Attribute Name	Attribute Description
paymentsDtls	Displays the payment details.
paymentEventId	Displays the payment event Id.
paymentId	Displays the payment Id.
messageValue	Displays the message returned by the create payment business service.
paymentStatusDesc	Indicates the reason why the create payment process is rejected.
statusCd	Indicates whether the payment creation request is successfully processed or not. The valid values are: • Success • Fail
paymentStatus	Indicates the payment status. The valid values are: • Frozen • Error

Copyright © 2024, Oracle and/or its affiliates.

Related Topics

For more information on...	See...
Sample Payment Creation Request and Response in JSON Format	Sample Payment Transfer Request and Response in JSON Format on page 442

Sample Payment Creation Request and Response in XML Format

Using the below example, you can create a payment request for the **0014198656** account

Sample Payment Creation Request in XML Format

```
<C1-createPayment>
    <accountNumberPayee>SAV12544785</accountNumberPayee>
    <accountNumberTypeCdPayee>CITIACCT</accountNumberTypeCdPayee>
  <divisionCodePayee>CA</divisionCodePayee>
  <matchType>SA-ID</matchType>
    <matchTypeValue>0077507587</matchTypeValue>
    <statementId></statementId>
    <currencyCd>USD</currencyCd>
    <accountingDt>2020-08-07</accountingDt>
    <tenderList>
        <tenderType>CASH</tenderType>
        <tenderCurrency>USD</tenderCurrency>
        <tenderAmount>250</tenderAmount>
        <checkNumber></checkNumber>
        <payorAccount>0014198656</payorAccount>
        <micrId>1111</micrId>
        <externalReferenceId>R63405610000000000000</externalReferenceId>
        <name>Tender 1</name>
        <paymentTenderCharacteristic>
           <characteristicType>C1-TAUTH</characteristicType>
           <characteristicValue>NOC</characteristicValue>
        </paymentTenderCharacteristic>
        <paymentTenderCharacteristic>
           <characteristicType>C1-TMPL</characteristicType>
           <characteristicValue>BS-200</characteristicValue>
        </paymentTenderCharacteristic>
    </tenderList>
    <tenderList>
        <tenderType>CHECK</tenderType>
        <tenderCurrency>USD</tenderCurrency>
        <tenderAmount>250</tenderAmount>
        <checkNumber>762998</checkNumber>
        <payorAccount>7374519014</payorAccount>
        <micrId>ANV</micrId>
        <externalReferenceId>MMAU-541-R</externalReferenceId>
        <name>Tender 2</name>
        <paymentTenderCharacteristic>
            <characteristicType>C1-TMPL</characteristicType>
            <characteristicValue>BS-200</characteristicValue>
         </paymentTenderCharacteristic>
         <paymentTenderCharacteristic>
            <characteristicType>C1-TAUTH</characteristicType>
            <characteristicValue>NOC</characteristicValue>
        </paymentTenderCharacteristic>
    </tenderList>
    <paymentCharacteristic>
        <characteristicType>ZZFK</characteristicType>
        <characteristicValue>5502393551</characteristicValue>
    </paymentCharacteristic>
    <paymentCharacteristic>
```

Copyright © 2024, Oracle and/or its affiliates.

```
        <characteristicType>ZZFK</characteristicType>
        <characteristicValue>5502393551</characteristicValue>
    </paymentCharacteristic>
</C1-createPayment>
```


Sample Payment Creation Response in XML Format


```
<C1-createPayment>
    <payeeAccount>3532174219</payeeAccount>
    <matchType>SA-ID</matchType>
    <matchTypeValue>3532174909</matchTypeValue>
    <statementId></statementId>
    <tenderList>
        <tenderType>CASH</tenderType>
        <tenderCurrency>USD</tenderCurrency>
        <tenderAmount>15.00</tenderAmount>
        <checkNumber></checkNumber>
        <payorAccount>0014198656</payorAccount>
        <micrId>1111</micrId>
        <externalReferenceId>R63405610000000000000</externalReferenceId>
        <name>Tender 1</name>
        <paymentTenderCharacteristic>
           <characteristicType>C1-TAUTH</characteristicType>
           <characteristicValue>NOC</characteristicValue>
        </paymentTenderCharacteristic>
        <paymentTenderCharacteristic>
           <characteristicType>C1-AOUTM</characteristicType>
           <characteristicValue>601169883875</characteristicValue>
        </paymentTenderCharacteristic>
    </tenderList>
    <tenderList>
        <tenderType>CHECK</tenderType>
        <tenderCurrency>USD</tenderCurrency>
        <tenderAmount>30.00</tenderAmount>
        <checkNumber>762998</checkNumber>
        <payorAccount>7374519014</payorAccount>
        <micrId>ANV</micrId>
        <externalReferenceId>MMAU-541-R</externalReferenceId>
        <name>Tender 2</name>
        <paymentTenderCharacteristic>
            <characteristicType>C1-TMPL</characteristicType>
            <characteristicValue>BS-200</characteristicValue>
        </paymentTenderCharacteristic>
        <paymentTenderCharacteristic>
            <characteristicType>C1-TAUTH</characteristicType>
            <characteristicValue>T13435465465</characteristicValue>
        </paymentTenderCharacteristic>
    </tenderList>
    <paymentCharacteristic>
        <characteristicType>ZZFK</characteristicType>
        <characteristicValue>5502393551</characteristicValue>
    </paymentCharacteristic>
    <paymentCharacteristic>
        <characteristicType>C1-REFNM</characteristicType>
        <characteristicValue>1237</characteristicValue>
    </paymentCharacteristic>
    <paymentsDtls>
        <paymentEventId>001804122299</paymentEventId>
        <paymentId>001535380932</paymentId>
        <messageValue>SUCCESS</messageValue>
        <paymentStatusDesc>Frozen</paymentStatusDesc>
        <paymentStatus>50</paymentStatus>
    </paymentsDtls>
```


Copyright © 2024, Oracle and/or its affiliates.

```
</C1-createPayment>
```

Note: We have given the above request and response samples to illustrate how the tags should be nested while sending a payment creation request from the external system. However, you should not seamlessly use this request sample in your environment because the required pre-requisite data may not be available in your environment. In such case, the system will give erroneous results.

Sample Payment Creation Request and Response in JSON Format

Using the below example, you can create a payment request for the 9663708377 account

Sample Payment Creation Request in XML Format

```
{
    -"C1-CreatePaymentService":
   {
        -"accountNumberPayee": -"SAV12544785",
        -"accountNumberTypeCdPayee": -"C1_F_ANO",
        -"divisionCodePayee": -"930",
        -"matchType": -"SA-ID",
        -"matchTypeValue": -"3583079049",
        -"statementId": -"",
        -"currencyCd": -"USD",
        -"paymentDt": -"2019-03-04",
        -"sourceSystem": -"",
        -"externalTransactionId": -"",
        -"externalSourceId": -"",
        -"tenderList":
      [
            {
                -"tenderType": -"CASH",
                -"tenderCurrency": -"USD",
                -"tenderAmount": 30,
                -"checkNumber": -"",
                -"payorAccount": -"9663708377",
                -"accountNumber": -"",
                -"accountNumberTypeCd": -"",
                -"divisionCode": -"",
                -"micrId": -"535343",
                -"externalReferenceId": -"PQWERT",
                -"name": -"",
                -"paymentTenderCharacteristic":
             [
                    {
                       -"characteristicType": -"C1-TAUTH",
                        -"characteristicValue": -"123"
                  -}
                -],
              -"paymentTenderCharacteristic":
             [
                    {
                        -"characteristicType": -"C1-REFNM",
                        -"characteristicValue": -"45"
                    -}
                -]
            -}
        -],
       -"tenderList":
      [
            {
                -"tenderType": -"CHECK",
                -"tenderCurrency": -"USD",
```

Copyright © 2024, Oracle and/or its affiliates.

```
                -"tenderAmount": 400,
                -"checkNumber": -"48112609",
                -"payorAccount": -"9663708377",
                -"accountNumber": -"",
                -"accountNumberTypeCd": -"",
                -"divisionCode": -"",
                -"micrId": -"9292",
                -"externalReferenceId": -"PQWERT",
                -"name": -"",
                -"paymentTenderCharacteristic":
            [
                    {
                        -"characteristicType": -"C1-TRVSE",
                        -"characteristicValue": -"ABC"
                    -}
                -]
            -}
        -],
        -"paymentCharacteristic":
      [
            {
                -"characteristicType": -"C1-REFNM",
                -"characteristicValue": -"12"
            -}
        -],
        -"paymentCharacteristic":
       [
            {
                -"characteristicType": -"C1RMPYMT",
               -"characteristicValue": -"600"
           -}
        -]
    -}
}
```


Sample Payment Creation Response in XML Format


```
{
    -"C1-CreatePaymentService":
   {
        -"accountNumberPayee": -"SAV12544785",
        -"accountNumberTypeCdPayee": -"C1_F_ANO",
        -"divisionCodePayee": -"930",
        -"matchType": -"SA-ID",
        -"matchTypeValue": -"3583079049",
        -"statementId": -"",
        -"currencyCd": -"USD",
        -"paymentDt": -"2019-03-04",
        -"tenderList":
      [
            {
                -"tenderType": -"CASH",
                -"tenderCurrency": -"USD",
                -"tenderAmount": 30.00,
                -"checkNumber": -"",
                -"payorAccount": -"9663708377",
                -"micrId": -"535343",
                -"externalReferenceId": -"PQWERT",
                -"paymentTenderCharacteristic":
            [
                    {
                       -"characteristicType": -"C1-TAUTH",
                        -"characteristicValue": -"123"
```


Copyright © 2024, Oracle and/or its affiliates.

```
                  -}
                -],
               -"paymentTenderCharacteristic":
              [
                     {
                         -"characteristicType": -"C1-REFNM",
                         -"characteristicValue": -"45"
                     -}
                -]
            -}
        -],
       -"tenderList":
      [
            {
                -"tenderType": -"CHECK",
                -"tenderCurrency": -"USD",
                -"tenderAmount": 400,
                -"checkNumber": -"48112609",
                -"payorAccount": -"9663708377",
                -"micrId": -"9292",
                -"externalReferenceId": -"PQWERT",
                -"paymentTenderCharacteristic":
              [
                     {
                         -"characteristicType": -"C1-TRVSE",
                         -"characteristicValue": -"ABC"
                     -}
                -]
            -}
        -],
        -"paymentCharacteristic":
      [
            {
                -"characteristicType": -"C1-REFNM",
                -"characteristicValue": -"12"
            -}
        -],
        -"paymentCharacteristic":
       [
            {
                -"characteristicType": -"C1RMPYMT",
               -"characteristicValue": -"600"
           -}
        -]
        -"paymentsDtls":
        {
            -"paymentEventId": -"966726526893",
            -"paymentId": -"966611828652",
            -"messageValue": -"SUCCESS",
            -"paymentStatusDesc": -"Frozen",
            -"paymentStatus": -"50"
        -}
    -}
}
```


Note: We have given the above request and response samples to illustrate how the attribute value pairs should be structured while sending a payment creation request from the external system. However, you should not seamlessly use this request sample in your environment because the required pre-requisite data may not be available in your environment. In such case, the system will give erroneous results.

Copyright © 2024, Oracle and/or its affiliates.

C1-CancelPayment Business Service

The **Cancel Payments (C1-CancelPaymentService)** business service is used for canceling the payments in the Oracle Revenue Management and Billing application based on the data provided from the external systems through an inbound web service.

You must provide the cancelation type and cancel reason as mandatory input parameters to the cancel payment business service. If type of cancelation type is **Tender Cancel**, you must provide the payment event Id, where all the frozen payments and tenders (belonging to this payment event Id) are canceled. If the type of cancelation is **Payment Cancel**, you must provide the payment Id, where all the frozen payments (belonging to this payment Id) are canceled.

Once the payment and tenders are successfully canceled, the business service responds by returning the following fields:

- Payment Event ID / Payment ID
- Response Message

Payment Cancellation Request in XML Format

The following code snippet illustrates how the tags should be nested while receiving the payment cancellation request in the XML format:

```
<c1-cancelPayment>
    <cancelReason></cancelReason>
    <cancelTypeFlag></cancelTypeFlag>
    <paymentEventId></paymentEventId>
    <paymentId></paymentId>
    <bank></bank>
    <bankAccount></bankAccount>
    <sourceSystem></sourceSystem>
    <externalTransactionId></externalTransactionId>
    <externalSourceId></externalSourceId>
</c1-cancelPayment>
```

Payment Cancellation Response in XML Format

The following code snippet illustrates how the tags are nested in the cancel payment business service schema:

```
<c1-cancelPayment>
    <cancelReason></cancelReason>
    <cancelTypeFlag></cancelTypeFlag>
    <paymentEventId></paymentEventId>
    <paymentId></paymentId>
    <bank></bank>
    <bankAccount></bankAccount>
    <sourceSystem></sourceSystem>
    <externalTransactionId></externalTransactionId>
    <externalSourceId></externalSourceId>
    <paymentsDtls>
        <paymentEventId></paymentEventId>
       <paymentId></paymentId>
      <paymentStatus></paymentStatus>
      <messageValue></messageValue>
      <statusCd></statusCd>
      <paymentStatusDesc></paymentStatusDesc>
    </paymentsDtls>
</c1-cancelPayment>
```

Copyright © 2024, Oracle and/or its affiliates.

Payment Cancellation Request in JSON Format

The following code snippet illustrates how the attribute/value pairs should be structured while receiving the payment cancellation request in the JSON format:

```
{
    -"C1-CancelPayment":
    {
        -"cancelReason": -""
        -"cancelTypeFlag": -"",
        -"paymentEventId": -"",
        -"paymentId": -"",
        -"bank": -"",
        -"bankAccount": -"",
        -"sourceSystem": -"",
        -"externalTransactionId": -"",
        -"externalSourceId": -"",
    -}
}
```

Payment Cancellation Response in JSON Format

The following code snippet illustrates how the attribute/value pairs are structured while sending the payment cancellation response in the JSON format:

```
{
    -"C1-CancelPayment":
    {
        -"cancelReason": -"",
        -"cancelTypeFlag": -"",
        -"paymentEventId": -"",
        -"paymentId": -"",
        -"bank": -"",
        -"bankAccount": -"",
        -"sourceSystem": -"",
        -"externalTransactionId": -"",
        -"externalSourceId": -"",
        -"paymentsDtls":
        {
            -"paymentEventId": -"",
            -"paymentId": -"",
            -"paymentStatus": -"",
            -"messageValue": -"",
            -"statusCd": -"",
            -"paymentStatusDesc": -""
        -}
    -}
}
```

Payment Cancellation Request and Response - Tags in XML Format

Note: We recommend you to refer the Payment Cancellation Request in XML Format on page 431 and Payment Cancellation Response in XML Format on page 431 topics in parallel while understanding the below mentioned tags. This will help you to understand how the tags are nested in the XML format.

Before calling the **C1-CancelPaymentService** business service through an inbound web service, you need to ensure that the cancel payment service contains the following tags:

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Tag Description	Mandatory (Yes or No)
cancelReason	Used to specify the reason for canceling the payments and tenders.	Yes
cancelTypeFlag	Used to indicate the type of cancelation. The valid values are: • Tender Cancel • Payment Cancel	Yes
paymentEventId	Used to specify the payment event id using which you want to cancel the payments and tenders.	Yes (Conditional) **Note:** This data is required if you have selected the type of cancelation as **Tender Cancel**.
paymentId	Used to specify the payment Id using which you want to cancel the payment.	Yes (Conditional) **Note:** This data is required while canceling the payment.
bank	Used to indicate the bank which should be charged for the cancellation of the payment.	No
bankAccount	Used to indicate the bank account which should be charged for the cancellation	No
sourceSystem	Used to indicate the external system from where the create payment request is received.	No
externalTransactionId	Used to indicate the external transaction Id of the payment on the payment upload interface.	No
externalSourceId	Used to specify the external source system Id.	No

The following table lists and describes the tags which appear in the payment cancellation response in the XML format:

Tag	Tag Description
paymentsDtls	Displays the payment details.
paymentEventId	Displays the payment event Id.
paymentId	Displays the payment Id.
paymentStatus	Indicates the status of the payment. The valid values are: • Frozen • Error
messageValue	Indicates an error that occurred while canceling the payment. **Note:** This tag appears when the value in the **statusCd** tag is set to **Fail**.
statusCd	Indicates whether the payment cancellation request is successfully processed or not. The valid values are: • Success • Fail

Copyright © 2024, Oracle and/or its affiliates.

Tag	Tag Description
paymentStatusDesc	Indicates the reason why the cancel payment operation is rejected.

Related Topics

For more information on...	See...
Sample Payment Cancellation Request and Response in XML Format	Sample Payment Cancellation Request and Response in XML Format on page 435

Payment Cancellation Request and Response - Tags in JSON Format

Note: We recommend you to refer the Payment Cancellation Request in JSON Format on page 432 and Payment Cancellation Response in JSON Format on page 432 topics in parallel while understanding the below mentioned attributes. This will help you to understand how the attributes are nested in the XML format.

Before calling the **C1-CancelPaymentService** business service, you need to ensure that the cancel payment service contains the following attributes:

Attribute Name	Attribute Description	Mandatory (Yes or No)
cancelReason	Used to specify the reason for canceling the payments and tenders.	Yes
cancelTypeFlag	Used to indicate the type of cancelation. The valid values are: • Tender Cancel • Payment Cancel	Yes
paymentEventId	Used to specify the payment event id using which you want to cancel the payments and tenders.	Yes (Conditional) **Note:** This data is required if you have selected the type of cancelation as **Tender Cancel**.
paymentId	Used to specify the payment Id using which you want to cancel the payment.	Yes (Conditional) **Note:** This data is required while canceling the payment.
bank	Used to indicate the bank which should be charged for the cancellation of the payment.	No
bankAccount	Used to indicate the bank account which should be charged for the cancellation	No
sourceSystem	Used to indicate the external system from where the create payment request is received.	No
externalTransactionId	Used to indicate the external transaction Id of the payment on the payment upload interface.	No
externalSourceId	Used to specify the external source system Id.	No

Copyright © 2024, Oracle and/or its affiliates.

The following table provides information about the response attribute in the cancel payment business service:

Attribute Name	Attribute Description
paymentsDtls	Displays the payment details.
paymentEventId	Displays the payment event Id.
paymentId	Displays the payment Id.
paymentStatus	Indicates the status returned by the cancel payment business service. The valid values are: • Frozen • Error
messageValue	Indicates an error that occurred while canceling the payment. **Note:** This tag appears when the value in the **statusCd** tag is set to **Fail**.
statusCd	Indicates whether the payment cancellation request is successfully processed or not. The valid values are: • Success • Fail
paymentStatusDesc	Indicates the reason why the cancel payment is rejected.

Related Topics

For more information on...	See...
Sample Payment Cancellation Request and Response in JSON Format	Sample Payment Cancellation Request and Response in JSON Format on page 436

Sample Payment Cancellation Request and Response in XML Format

Using the below example, you can cancel a payment request for the 438418685299 payment event.

Sample Payment Cancellation Request in XML Format

```
<c1-cancelPayment>
    <cancelReason>ENTR</cancelReason>
    <cancelTypeFlag>T</cancelTypeFlag>
    <paymentEventId>438418685299</paymentEventId>
    <paymentId></paymentId>
    <bank>BOA</bank>
    <bankAccount>BOA</bankAccount>
    <sourceSystem>ZZ_IWS_ES</sourceSystem>
    <externalTransactionId>122</externalTransactionId>
    <externalSourceId>444</externalSourceId>
</c1-cancelPayment>
```

Sample Payment Cancellation Response in XML Format

```
<c1-cancelPayment>
    <cancelReason>ENTR</cancelReason>
    <cancelTypeFlag>T</cancelTypeFlag>
    <paymentEventId>438418685299</paymentEventId>
    <paymentId></paymentId>
    <bank>BOA</bank>
```

Copyright © 2024, Oracle and/or its affiliates.

```
    <bankAccount>BOA</bankAccount>
    <sourceSystem>ZZ_IWS_ES</sourceSystem>
    <externalTransactionId>122</externalTransactionId>
    <externalSourceId>444</externalSourceId>
    <paymentsDtls>
        <paymentEventId>438418685299</paymentEventId>
       <paymentId></paymentId>
      <paymentStatus>Frozen</paymentStatus>
      <messageValue></messageValue>
      <statusCd>Success</statusCd>
      <paymentStatusDesc></paymentStatusDesc>
    </paymentsDtls>
</c1-cancelPayment>
```

Note: We have given the above request and response samples to illustrate how the tags should be nested while sending a payment cancellation request from the external system. However, you should not seamlessly use this request sample in your environment because the required pre-requisite data may not be available in your environment. In such case, the system will give erroneous results.

Sample Payment Cancellation Request and Response in JSON Format

Using the below example, you can cancel a payment request for the 438418685299 payment.

Sample Payment Cancellation Request in JSON Format

```
{
    -"C1-CancelPayment":
    {
        -"cancelReason": -"ENTR"
        -"cancelTypeFlag": -"T",
        -"paymentEventId": -"",
        -"paymentId": -"547019197259",
        -"bank": -"BOA",
        -"bankAccount": -"BOA",
        -"sourceSystem": -"ZZ_IWS_ES",
        -"externalTransactionId": -"122",
        -"externalSourceId": -"444",
    -}
}
```

Sample Payment Cancellation Response in JSON Format

```
{
    -"C1-CancelPayment":
    {
        -"cancelReason": -"ENTR"
        -"cancelTypeFlag": -"T",
        -"paymentEventId": -"",
        -"paymentId": -"547019197259",
        -"bank": -"BOA",
        -"bankAccount": -"BOA",
        -"sourceSystem": -"ZZ_IWS_ES",
        -"externalTransactionId": -"122",
        -"externalSourceId": -"444",
        -"paymentsDtls":
        {
            -"paymentEventId": -"",
            -"paymentId": -"547019197259",
            -"paymentStatus": -"Frozen",
            -"messageValue": -"",
            -"statusCd": -"Success",
            -"paymentStatusDesc": -""
```

Copyright © 2024, Oracle and/or its affiliates.

```
            -}
        -}
}
```

Note: We have given the above request and response samples to illustrate how the attribute value pairs should be structured while sending a payment cancellation request from the external system. However, you should not seamlessly use this request sample in your environment because the required pre-requisite data may not be available in your environment. In such case, the system will give erroneous results.

C1-TransferPayment Business Service

The **Transfer Payments (C1-TransferPaymentService)** business service is used to transfer the frozen payments in the Oracle Revenue Management and Billing application based match value which is provided from the external system through an inbound web service.

You must provide the transfer type and transfer reason as mandatory input parameters to the transfer payment business service. If type of transfer is **Payment Event**, you must provide the payment event Id, where all the frozen payments (belonging to this payment event Id) are canceled, and a new payment in the frozen state is created. The summation of all these payment amount is used as the transfer amount, applied to the provided match value. If the type of cancelation is **Payment ID**, you must provide the payment Id, where the payment amount is used as the transfer amount, applied to the provided match value.

Once the payments are successfully transferred, the business service responds by returning the following fields:

- Payment Payment ID : {new payment created}
- Response Message

Payment Transfer Request in XML Format

The following code snippet illustrates how the tags should be nested while receiving the payment transfer request in the XML format:

```
<c1-transferPayment>
    <paymentId></paymentId>
    <paymentEventId></paymentEventId>
    <transferType></transferType>
    <transferReason></transferReason>
    <matchType></matchType>
    <matchTypeValue></matchTypeValue>
    <transferAmount></transferAmount>
</c1-transferPayment>
```

Payment Transfer Response in XML Format

The following code snippet illustrates how the tags should be nested while sending the payment transfer response in the XML format:

```
<c1-transferPayment>
    <paymentId></paymentId>
    <paymentEventId></paymentEventId>
    <transferType></transferType>
    <transferReason></transferReason>
    <matchType></matchType>
    <matchTypeValue></matchTypeValue>
    <transferAmount></transferAmount>
    <paymentResponse>
        <paymentEventId></paymentEventId>
        <paymentId></paymentId>
```

Copyright © 2024, Oracle and/or its affiliates.

```
        <paymentStatus></paymentStatus>
        <messageValue></messageValue>
        <paymentStatusDesc></paymentStatusDesc>
    </paymentResponse>
</c1-transferPayment>
```

Payment Transfer Request in JSON Format

The following code snippet illustrates how the tags should be nested while receiving the payment transfer request in the JSON format:

```
{
    -"c1-transferPayment":
    {
        -"paymentId": -"",
        -"paymentEventId": -"",
        -"transferType": -""
        -"transferReason": -""
        -"matchType": -""
        -"transferAmount": -""
    -}
}
```

Payment Transfer Response in JSON Format

The following code snippet illustrates how the tags should be nested while sending the payment transfer response in the JSON format:

```
{
    -"c1-transferPayment":
    {
        -"paymentId": -"",
        -"paymentEventId": -"",
        -"transferType": -"",
        -"transferReason": -"",
        -"matchType": -"",
        -"transferAmount": -""
        -"paymentResponse":
        {
            -"paymentEventId": -"",
            -"paymentId": -"",
            -"paymentStatus": -"",
            -"messageValue": -"",
            -"paymentStatusDesc": -""
        -}
    -}
}
```

Payment Transfer Request and Response - Tags in XML Format

Note: We recommend you to refer the Payment Transfer Request in XML Format on page 437 and Payment Transfer Response in XML Format on page 437 topics in parallel while understanding the below mentioned tags. This will help you to understand how the tags are nested in the XML format.

Before calling the **C1-TransferPaymentService** business service through an inbound web service, you need to ensure that the payment transfer service contains the following tags:

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Tag Description	Mandatory (Yes or No)
paymentId	Used to specify the payment Id using which you want to transfer the payment.	Yes (Conditional) **Note:** This data is required when the transfer payment process is initiated.
paymentEventId	Used to specify the payment event Id using which you want to transfer all the frozen payments to the specified match value.	Yes (Conditional) **Note:** This data is required when the transfer payment process is initiated.
transferType	Used to indicate the type of payment transfer. The valid values are: • Payment Event (E) • Payment (P) **Note:** You must select a transfer type.	Yes
transferReason	Used to specify the reason to perform the payment transfer. **Note:** You must specify the transfer reason which is already defined in the system.	Yes
matchType	Used to indicate the match type using which you want to transfer the payment. **Note:** You must specify the match type which is already defined in the system.	Yes
matchTypeValue	Used to specify a particular entity (such as bill ID or contract ID) for the match type. **Note:** The match value must be a valid bill ID or contract ID.	Yes
sourceSystem	Used to indicate the external system from where the create payment request is received.	No
externalTransactionId	Used to indicate the external transaction Id of the payment on the payment upload interface.	No
externalSourceId	Used to specify the external source system Id.	No
transferAmount	Used to specify the amount to perform the payment transfer.	No

The following table lists and describes the tags which appear in the payment transfer response in the XML format:

Tag Name	Tag Description
paymentsDtls	Displays the payment details.
paymentEventId	Displays the payment event Id.

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Tag Description
messageValue	Indicates an error that occurred while transferring the payment. **Note:** This tag appears when the value in the **statusCd** tag is set to **Fail**.
paymentStatus	Indicates the status of the payment. The valid values are: • Frozen • Error
statusCd	Indicates whether the payment transfer request is successfully processed or not. The valid values are: • Success • Fail
paymentStatusDesc	Indicates the reason why the cancel payment operation is rejected.

Related Topics

For more information on...	See...
Sample Payment Transfer Request and Response in XML Format	Sample Payment Transfer Request and Response in XML Format on page 442

Payment Transfer Request and Response - Attribute/Value Pairs in JSON Format

Note: We recommend you to refer the Payment Transfer Request in JSON Format on page 438 and Payment Transfer Response in JSON Format on page 438 topics in parallel while understanding the below mentioned attributes. This will help you to understand how the attributes are nested in the XML format.

Before calling the **C1-TransferPaymentService** business service through an inbound web service, you need to ensure that the payment transfer service contains the following attributes:

Attribute Name	Attribute Description	Mandatory (Yes or No)
paymentId	Used to specify the payment Id using which you want to transfer the payment.	Yes (Conditional) **Note:** This data is required when the transfer payment process is initiated.
paymentEventId	Used to specify the payment event Id using which you want to transfer all the frozen payments to the specified match value.	Yes (Conditional) **Note:** This data is required when the transfer payment process is initiated.
transferType	Used to indicate the type of payment transfer. The valid values are: • Payment Event (E) • Payment (P) **Note:** You must select a transfer type which is already defined in the system.	Yes

Copyright © 2024, Oracle and/or its affiliates.

Attribute Name	Attribute Description	Mandatory (Yes or No)
transferReason	Used to specify the reason to perform the payment transfer. **Note:** You must specify the transfer reason which is already defined in the system.	Yes
matchType	Used to indicate the match type using which you want to transfer the payment. **Note:** You must specify the match type which is already defined in the system.	Yes
matchTypeValue	Used to specify a particular entity (such as bill ID or contract ID) for the match type. **Note:** The match value must be a valid bill ID or contract ID.	Yes
sourceSystem	Used to indicate the external system from where the create payment request is received.	No
externalTransactionId	Used to indicate the external transaction Id of the payment on the payment upload interface.	No
externalSourceId	Used to specify the external source system Id.	No
transferAmount	Used to specify the amount to perform the payment transfer.	No

The following table lists and describes the attributes which appear in the payment transfer response in the XML format:

Attribute Name	Attribute Description
paymentsDtls	Displays the payment details.
paymentEventId	Displays the payment event Id.
messageValue	Indicates an error that occurred while transferring the payment. **Note:** This attribute appears when the value in the **statusCd** tag is set to **Fail**.
paymentStatus	Indicates the status of the payment. The valid values are: • Frozen • Error
statusCd	Indicates whether the payment transfer request is successfully processed or not. The valid values are: • Success • Fail
paymentStatusDesc	Indicates the reason why the payment transfer operation is rejected.

Related Topics

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
Sample Payment Transfer Request and Response in JSON Format	Sample Payment Transfer Request and Response in JSON Format on page 442

Sample Payment Transfer Request and Response in XML Format

Using the below example, you can transfer a payment request for the 001878288481 payment event.

Sample Payment Transfer Request in XML Format

```
<c1-transferPayment>
    <paymentId></paymentId>
    <paymentEventId>001878288481</paymentEventId>
    <transferType>E</transferType>
    <transferReason>ENTR</transferReason>
    <matchType>SA-ID</matchType>
    <matchTypeValue>5571092420</matchTypeValue>
    <transferAmount>200</transferAmount>
</c1-transferPayment>
```

Sample Payment Transfer Response in XML Format

```
<c1-transferPayment>
    <paymentId></paymentId>
    <paymentEventId>001878288481</paymentEventId>
    <transferType>E</transferType>
    <transferReason>ENTR</transferReason>
    <matchType>SA-ID</matchType>
    <matchTypeValue>5571092420</matchTypeValue>
    <transferAmount>200</transferAmount>
    <paymentResponse>
        <paymentId></paymentId>
        <paymentEventId>001878288481</paymentEventId>
        <paymentStatus>Frozen</paymentStatus>
        <messageValue></messageValue>
        <statusCd>Success</statusCd>
        <paymentStatusDesc></paymentStatusDesc>
    </paymentResponse>
</c1-transferPayment>
```

Note: We have given the above request and response samples to illustrate how the tags should be nested while sending a payment transfer request from the external system. However, you should not seamlessly use this request sample in your environment because the required pre-requisite data may not be available in your environment. In such case, the system will give erroneous results.

Sample Payment Transfer Request and Response in JSON Format

Using the below example, you can transfer a payment request for the 547019197259 payment.

Sample Payment Cancellation Request in JSON Format

```
{
    -"C1-TransferPayment":
    {
        -"paymentId": -"547019197259"
        -"paymentEventId": -"",
        -"transferType": -"E",
        -"paymentId": -"",
        -"transferReason": -"ENTR",
        -"matchType": -"SA-ID",
```

Copyright © 2024, Oracle and/or its affiliates.

```
        -"matchTypeValue": -"5571092420",
        -"transferAmount": -"200"
    -}
}
```

Sample Payment Cancellation Response in JSON Format

```
{
    -"C1-TransferPayment":
    {
        -"paymentId": -"547019197259"
        -"paymentEventId": -"",
        -"transferType": -"E",
        -"paymentId": -"",
        -"transferReason": -"ENTR",
        -"matchType": -"SA-ID",
        -"matchTypeValue": -"5571092420",
        -"transferAmount": -"200"
        {
            -"paymentId": -"547019197259",
            -"paymentEventId": -""
            -"paymentStatus": -"Frozen",
            -"messageValue": -"",
            -"statusCd": -"Success",
            -"paymentStatusDesc": -""
        -}
    -}
}
```

Note: We have given the above request and response samples to illustrate how the attribute value pairs should be structured while sending a payment transfer request from the external system. However, you should not seamlessly use this request sample in your environment because the required pre-requisite data may not be available in your environment. In such case, the system will give erroneous results.

C1-ViewPayment Business Service

The **View Payments (C1-ViewPaymentService)** business service is used for viewing the payment history for an account in the Oracle Revenue Management and Billing application from the external systems through an inbound web service. This business service fetches the maximum of 50 records based on the payment date.

Note: The difference between **Payment Date From** and **Payment Date To** should not exceed more than a year.

The view payment business service responds by returning the following fields:

- Payment Event ID
- Payment ID
- Payment Date
- Payment Amount
- Payment Status
- Match Type
- Match Value
- Payment Currency
- Cancelation Reason

Copyright © 2024, Oracle and/or its affiliates.

Payment View Request in XML Format

The following code snippet illustrates how the tags should be nested while receiving the payment view request in the XML format:

```
<C1-viewPayment>
    <accountIdentifierType></accountIdentifierType>
    <accountNumber></accountNumber>
    <accountId></accountId>
    <tenderType></tenderType>
    <checkNumber></checkNumber>
  <division></division>
    <payDateFrom></payDateFrom>
    <paymentDateTo></paymentDateTo>
    <sourceSystem></sourceSystem>
    <c1ExternalTransactionId></c1ExternalTransactionId>
    <externalSourceId></externalSourceId>
 </c1-viewPayment>
```

Payment View Response in XML Format

The following code snippet illustrates how the tags should be nested while sending the payment view response in the XML format:

```
<C1-viewPayment>
    <accountIdentifierType></accountIdentifierType>
    <accountNumber></accountNumber>
    <accountId></accountId>
    <tenderType></tenderType>
    <checkNumber></checkNumber>
  <division></division>
    <payDateFrom></payDateFrom>
    <paymentDateTo></paymentDateTo>
    <sourceSystem></sourceSystem>
    <c1ExternalTransactionId></c1ExternalTransactionId>
    <externalSourceId></externalSourceId>
    <getPayHistResponse>
        <statusCd>Success</statusCd>
        <payHistList>
            <paymentEventId></paymentEventId>
            <paymentId></paymentId>
            <paymentDate></paymentDate>
            <paymentAmount></paymentAmount>
            <currency></currency>
            <matchTypeDescription></matchTypeDescription>
            <paymentStatusDesc></paymentStatusDesc>
        </payHistList>
    </getPayHistResponse>
</c1-viewPayment>
```

Payment View Request in JSON Format

The following code snippet illustrates how the tags should be nested while receiving the payment view request in the JSON format:

```
{
    -"C1-ViewPaymentService":
    {
       -"accountIdentifierType": -"",
       -"accountNumber": -"",
       -"accountId": -"",
```

Copyright © 2024, Oracle and/or its affiliates.

```
        -"tenderType": -"",
        -"checkNumber": -"",
        -"division": -"",
        -"sourceSystem": -"",
        -"c1ExternalTransactionId": -"",
        -"externalSourceId": -""
   -}
}
```

Payment View Response in JSON Format

The following code snippet illustrates how the tags should be nested while sending the payment view response in the JSON format:

```
{
    -"C1-ViewPaymentService":
    {
       -"accountIdentifierType": -"",
       -"accountNumber": -"",
       -"accountId": -"",
       -"tenderType": -"",
       -"checkNumber": -"",
       -"division": -"",
       -"sourceSystem": -"",
       -"c1ExternalTransactionId": -"",
       -"externalSourceId": -"",
       -"getPayHistResponse": -""
       {
           -"statusCd": -"",
           -"payHistList":
           [
               {
                   -"paymentEventId": -"",
                   -"paymentId": -"",
                   -"paymentDate": -"",
                   -"paymentAmount": -"",
                   -"currency": -"",
                   -"matchTypeDescription": -"",
                   -"paymentStatusDesc": -""
               -}
           -]
       -}
    -}
}
```

Payment View Request and Response - Tags in XML Format

Note: We recommend you to refer the Payment View Request in XML Format on page 444 and Payment View Response in XML Format on page 444 topics in parallel while understanding the below mentioned tags. This will help you to understand how the tags are nested in the XML format.

Before calling the **C1-ViewPaymentService** business service through an inbound web service, you need to ensure that the payment view request contains the following tags:

Tag Name	Tag Description	Mandatory (Yes or No)
accountIdentifierType	Used to specify the account identifier type.	Yes (Conditional)

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Tag Description	Mandatory (Yes or No)
		Note: This data is required while viewing the payment history for an account.
accountNumber	Used to specify the value for the account identifier type of the payee. **Note:** You must specify the account number which is already defined in the system.	Yes (Conditional) **Note:** This data is required if the **accountIdentifierType** is specified.
accountId	Used to indicate the account for which the adjustment history is retrieved. **Note:** You must specify an accountId which is already defined in the system.	Yes (Conditional) **Note:** This data is required if the **accountIdentifierType** and **accountNumber** is not specified.
tenderType	Used to indicate the tender type using which you want to view the payment for an account. **Note:** You must indicate the tender type which is already defined in the application.	No
checkNumber	Used to specify the check number, if payment is made through check.	No
division	Used to indicate the division code. **Note:** You must specify the division code which is already defined in the system.	Yes (Conditional) **Note:** This data is required while viewing the payment history for an account.
payDateFrom	Used to specify the date from when the payment is effective for an account. **Note:** The payDateFrom cannot not be later than the paymentDateTo.	No
paymentDateTo	Used to specify the date till when the payment is effective for an account. **Note:** The payDateFrom cannot not be earlier than the paymentDateTo.	No
sourceSystem	Used to indicate the external system from where the view payment request is received.	No
c1ExternalTransactionId	Used to indicate the external transaction Id of the payment.	No
externalSourceId	Used to specify the external source system Id.	No

The following table provides information about the response tags in the view payment business service:

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Tag Description
getPayHistResponse	Displays the response returned by the business service in relation to the payment history.
statusCd	Indicates whether the view payment request is successfully processed or not. The valid values are: • Success • Fail
messageText	Indicates the reason for an error occurrence.
payHistList	Displays the list of payment details for an account.
paymentEventId	Displays the payment event ID.
paymentId	Displays the payment ID whose payment history is viewed.
paymentDate	Displays the date when the payment was made.
paymentAmount	Displays the payment amount.
paymentStatus	Indicates the status of the payment. The valid values are: • Frozen • Error
matchType	Indicates the match type using which the payment history is matched.
matchValue	Indicates the entity (bill ID or contract ID) against which the payment history is matched.
currency	Indicates the currency in which the payment was made.
cancelReason	Indicates the reason why the payment was canceled.
matchTypeDescription	Indicates the description of the match type.
paymentStatusDesc	Indicates the description of the payment status.

Related Topics

For more information on...	See...
Sample Payment View Request and Response in XML Format	Payment View Request and Response - Tags in XML Format on page 445

Payment View Request and Response - Attribute/Value Pairs in JSON Format

Note: We recommend you to refer thePayment View Request in JSON Format on page 444 and Payment View Response in JSON Format on page 445 topics in parallel while understanding the below mentioned attributes. This will help you to understand how the attributes are nested in the XML format.

Before calling the **C1-ViewPaymentService** business service through an inbound web service, you need to ensure that the payment view request contains the following attributes:

Attribute Name	Attribute Description	Mandatory (Yes or No)
accountIdentifierType	Used to specify the account identifier type.	Yes (Conditional)

Copyright © 2024, Oracle and/or its affiliates.

Attribute Name	Attribute Description	Mandatory (Yes or No)
		Note: This data is required while viewing the payment history for an account.
accountNumber	Used to specify the value for the account identifier type of the payee. **Note:** You must specify the account number which is already defined in the system.	Yes (Conditional) **Note:** This data is required if the **accountIdentifierType** is specified.
accountId	Used to indicate the account for which the adjustment history is retrieved. **Note:** You must specify an accountId which is already defined in the system.	Yes (Conditional) **Note:** This data is required if the **accountIdentifierType** and **accountNumber** is not specified.
tenderType	Used to indicate the tender type using which you want to view the payment for an account. **Note:** You must indicate the tender type which is already defined in the application.	No
checkNumber	Used to specify the check number, if payment is made through check.	No
division	Used to indicate the division code. **Note:** You must specify the division code which is already defined in the system.	Yes (Conditional) **Note:** This data is required while viewing the payment history for an account.
payDateFrom	Used to specify the date from when the payment is effective for an account. **Note:** The payDateFrom cannot not be later than the paymentDateTo.	No
paymentDateTo	Used to specify the date till when the payment is effective for an account. **Note:** The payDateFrom cannot not be earlier than the paymentDateTo.	No
sourceSystem	Used to indicate the external system from where the view payment request is received.	No
c1ExternalTransactionId	Used to indicate the external transaction Id of the payment.	No
externalSourceId	Used to specify the external source system Id.	No

The following table lists and describes the attributes which appear in the payment view response in the JSON format:

Copyright © 2024, Oracle and/or its affiliates.

Attribute Name	Attribute Description
getPayHistResponse	Displays the response returned by the business service in relation to the payment history.
statusCd	Indicates whether the view payment request is successfully processed or not. The valid values are: • Success • Fail
messageText	Indicates the reason for an error occurrence.
payHistList	Displays the list of payment details for an account.
paymentEventId	Displays the payment event ID.
paymentId	Displays the payment ID whose payment history is viewed.
paymentDate	Displays the date when the payment was made.
paymentAmount	Displays the payment amount.
paymentStatus	Indicates the status of the payment. The valid values are: • Frozen • Error
matchType	Indicates the match type using which the payment history is matched.
matchValue	Indicates the entity (bill ID or contract ID) against which the payment history is matched.
currency	Indicates the currency in which the payment was made.
cancelReason	Indicates the reason why the payment was canceled.
matchTypeDescription	Indicates the description of the match type.
paymentStatusDesc	Indicates the description of the payment status.

Related Topics

For more information on...	See...
Sample Payment View Request and Response in JSON Format	Sample Payment View Request and Response in JSON Format on page 450

Sample Payment View Request and Response in XML Format

Using the below example, you can view the payment history of the 9007760606 account.

Sample Payment View Request in XML Format

```
<c1-viewPayment>
    <accountIdentifierType>CITIACCT</accountIdentifierType>
    <accountNumber>ZZZZACCT</accountNumber>
    <accountId>9007760606</accountId>
    <tenderType>CHECK</tenderType>
    <checkNumber>7005</checkNumber>
  <division>CA</division>
    <sourceSystem>ZZ_IWS_ES</sourceSystem>
    <c1ExternalTransactionId>122</c1ExternalTransactionId>
    <externalSourceId>444</externalSourceId>
</c1-viewPayment>
```

Copyright © 2024, Oracle and/or its affiliates.

Sample Payment View Response in XML Format

```
<c1-viewPayment>
    <accountIdentifierType>CITIACCT</accountIdentifierType>
    <accountNumber>ZZZZACCT</accountNumber>
    <accountId>9007760606</accountId>
    <tenderType>CHECK</tenderType>
    <checkNumber>7005</checkNumber>
    <division>CA</division>
    <sourceSystem>ZZ_IWS_ES</sourceSystem>
    <c1ExternalTransactionId>122</c1ExternalTransactionId>
    <externalSourceId>444</externalSourceId>
    <getPayHistResponse>
        <statusCd>Success</statusCd>
        <payHistList>
           <paymentEventId>001095181478</paymentEventId>
           <paymentId>001426421608</paymentId>
           <paymentDate>2020-08-19</paymentDate>
           <paymentAmount>500.00</paymentAmount>
           <currency>USD</currency>
           <matchTypeDescription>SA ID match</matchTypeDescription>
           <paymentStatusDesc>Error</paymentStatusDesc>
        </payHistList>
        <payHistList>
            <paymentEventId>001028557769</paymentEventId>
            <paymentId>001807743707</paymentId>
            <paymentDate>2020-08-17</paymentDate>
            <paymentAmount>1000.00</paymentAmount>
            <matchValue>0077507587</matchValue>
            <currency>USD</currency>
            <matchTypeDescription>SA ID match</matchTypeDescription>
            <paymentStatusDesc>Frozen</paymentStatusDesc>
        </payHistList>
        <payHistList>
           <paymentEventId>001059826039</paymentEventId>
           <paymentId>001007431496</paymentId>
           <paymentDate>2020-08-14</paymentDate>
           <paymentAmount>260.00</paymentAmount>
           <matchValue>0077507587</matchValue>
           <currency>USD</currency>
           <cancelReason>ENTR</cancelReason>
           <matchTypeDescription>SA ID match</matchTypeDescription>
           <paymentStatusDesc>Canceled</paymentStatusDesc>
        </payHistList>
    </getPayHistResponse>
 </c1-viewPayment>
```

Note: We have given the above request and response samples to illustrate how the tags should be nested while sending a payment view request from the external system. However, you should not seamlessly use this request sample in your environment because the required pre-requisite data may not be available in your environment. In such case, the system will give erroneous results.

Sample Payment View Request and Response in JSON Format

Using the below example, you can view the payment history of the 8007760655 account.

Sample Payment View Request in JSON Format

```
{
    -"C1-ViewPaymentService":
    {
        -"tenderType": -"CHEC",
```

Copyright © 2024, Oracle and/or its affiliates.

```
        -"accountIdentifierType": -"C1_F_ANO",
        -"accountNumber": -"SAV12544785",
        -"accountId": -"8007760655",
        -"tenderType": -"CHECK",
        -"division": -"CB",
        -"sourceSystem": -"ZZ_IWS_ES",
        -"c1ExternalTransactionId": -"122",
        -"externalSourceId": -"444",
    -}
}
```


Sample Payment View Response in JSON Format


```
{
    -"C1-ViewPaymentService":
    {
        -"getPayHistResponse":
        {
            -"statusCd": -"Success",
            -"payHistList": -
            [
                {
                    -"paymentEventId": -"966818046432",
                    -"paymentId": -"966504142391",
                    -"paymentDate": -"2020-08-13",
                    -"paymentAmount": -"410.00",
                    -"matchValue": -"3583079049",
                    -"currency": -"USD",
                    -"cancelReason": -"ENTR",
                    -"matchTypeDescription": -"Contract ID match",
                    -"paymentStatusDesc": -"Canceled"
                -},
                {
                    -"paymentEventId": -"966364498272",
                    -"paymentId": -"966635189345",
                    -"paymentDate": -"2020-08-13",
                    -"paymentAmount": -"400.00",
                    -"matchValue": -"3583079049",
                    -"currency": -"USD",
                    -"cancelReason": -"ENTR",
                    -"matchTypeDescription": -"Contract ID match",
                    -"paymentStatusDesc": -"Canceled"
                -},
                {
                    -"paymentEventId": -"966584017975",
                    -"paymentId": -"966265993623",
                    -"paymentDate": -"2020-08-12",
                    -"paymentAmount": -"410.00",
                    -"matchValue": -"3583079049",
                    -"currency": -"USD",
                    -"cancelReason": -"ENTR",
                    -"matchTypeDescription": -"Contract ID match",
                    -"paymentStatusDesc": -"Canceled"
                -},
                {
                    -"paymentEventId": -"966410695808",
                    -"paymentId": -"966550246069",
                    -"paymentDate": -"2020-08-06",
                    -"paymentAmount": -"37500.00",
                    -"matchValue": -"3583079049",
                    -"currency": -"USD",
                    -"matchTypeDescription": -"Contract ID match",
                    -"paymentStatusDesc": -"Frozen"
```


Copyright © 2024, Oracle and/or its affiliates.

```
            -},
            {
                  -"paymentEventId": -"966335717439",
                  -"paymentId": -"966824712954",
                  -"paymentDate": -"2020-08-06",
                  -"paymentAmount": -"37500.00",
                  -"matchValue": -"3583079049",
                  -"currency": -"USD",
                  -"matchTypeDescription": -"Contract ID match",
                  -"paymentStatusDesc": -"Frozen"
                -}
            -]
        -},
    -}
-}
```


Note: We have given the above request and response samples to illustrate how the attribute value pairs should be structured while sending a payment view request from the external system. However, you should not seamlessly use this request sample in your environment because the required pre-requisite data may not be available in your environment. In such case, the system will give erroneous results.

Chapter 4

Adjustments

Topics:

- The Big Picture Of Adjustments
- Maintaining Adjustments
- How and When To Use An Adjustment
- Interfacing Adjustments From External Sources
- Adjustment Creation, Cancellation, and View Through Inbound Web Service

In this section, we describe when and how to adjust a contract's balance.

The Big Picture Of Adjustments

The topics in this section provide background information about a variety of adjustment issues.

Fastpath: We strongly recommend familiarizing yourself with the topics described in The Financial Big Picture to fully appreciate the place of an adjustment in the system's financial architecture. In particular, refer to Setting Up Adjustment Types and Setting Up Adjustment Type Profiles.

Adjustments - Current Balance versus Payoff Balance

Adjusting how much a customer owes involves changing a contract's payoff balance and/or current balance by creating an adjustment. In this section we describe these two balances.

CAUTION: If you do not understand the difference between payoff balance and current balance, refer to Current Amount versus Payoff Amount.

When Current Balance Equals Payoff Balance

For most contracts, payoff balance and current balance are always the same (or in colloquial speech - the amount the customer thinks they owe equals what they really owe). In this situation, an adjustment is easy: both payoff balance and current balance are adjusted by the same value.

Let's run through a typical example. The values in the payoff balance and current balance columns reflect the amount due after the financial transaction has been applied (i.e., the running balance):

The following table illustrates an example where the payoff and current balances are same:

Date	Financial Transaction	Payoff Balance	Current Balance
1-Jan-99	Bill: $125	125	125
15-Jan-99	Payment: $150	-25	-25
2-Feb-99	Bill: $175	150	150
14-Feb-99	Payment: $150	0	0
3-Mar-99	Bill: $200	200	200
15-Mar-99	Payment: $150	50	50
2-Apr-99	Bill: $225	275	275
27-Apr-99	Adj: Late Payment Charge $10	285	285

As you can see, payoff balance and current balance are always in sync.

When Current Balance Differs From Payoff Balance

For some contracts, payoff balance and current balance differ (or in colloquial speech â€‘ the amount the customer thinks they owe differs from what they would owe if they wanted to payoff their account).

Let's run through an example of a customer on a budget to illustrate a classic contract where these two balances are not the same. The values in the payoff balance and current balance columns reflect the amount due after the financial transaction has been applied:

The following table illustrates an example where the payoff and current balances are different:

Copyright © 2024, Oracle and/or its affiliates.

Date	Financial Transaction	Payoff Balance	Current Balance
1-Jan-99	Bill: $125, Budget $150	125	150
15-Jan-99	Payment: $150	-25	0
2-Feb-99	Bill: $175, Budget $150	150	150
14-Feb-99	Payment: $150	0	0
3-Mar-99	Bill: $200, Budget $150	200	150
15-Mar-99	Payment: $150	50	0
2-Apr-99	Adj: Mar 3, 97 bill incorrect due to discrepancies; manual correction of -$33.	17	0

Notice the adjustment of -$33 affects only payoff amount. This is because the customer is on a budget. The adjustment will not affect their monthly payment and therefore it doesn't affect how much they think they owe.

The following table lists the various types of contracts where the payoff and current balances may differ:

Type Of Contract	What Payoff Balance Holds	What Current Balance Holds
Deposit	Payoff balance holds the amount of deposit you are holding on behalf of the customer's debt. This amount is showed as a credit (with a negative sign) because you truly owe the customer this money.	Current balance holds the amount the customer owes for a deposit.
Charitable contribution	N/A - charitable contribution contracts never have a payoff balance.	Current balance holds the amount the customer owes for a charitable contribution.
Payment arrangements	Payoff balance holds the total amount of delinquent debt that remains unpaid. When a payment arrangement is started, you transfer the payoff balance from the financial services contract to the payment arrangement contract. Each billing period, the customer pays down this balance.	Current balance holds the amount the customer owes in respect of the billed payment arrangement amounts.
Financial service agreements on a budget plan	Payoff balance holds the actual amount owed (actual bills minus payments).	Current balance holds the amount the customer owes in respect of the billed budget amount.
Loans	Payoff balance holds the payoff amount for the loan, which is the current balance (the billed loan amount) plus the principal balance (the unbilled loan amount) and any accrued interest charges.	Current balance holds the amount the customer owes in respect of the billed amount (the periodic payment amount).

Adjustment Type And Balances

When you create an adjustment, you must define its adjustment type. The adjustment type controls how payoff balance and current balance are affected by the adjustment amount.

You can only pick adjustment types that make sense for the contract. For example, if your organization doesn't allow budget billing, all adjustments for a financial service contract will affect payoff balance and current balance the same (you can't get the balances out-of-sync).

Copyright © 2024, Oracle and/or its affiliates.

The various types of adjustments that may be linked to a contract are controlled by the adjustment profiles defined on the contract's contract type.

Fastpath: Refer to Adjustment Type Controls Everything for more information.

Canceling Adjustments

You may cancel any frozen adjustment. Canceling the adjustment creates another financial transaction that reverses the financial effects of the original adjustment. The impact of the cancellation appears on the customer's next bill. You may view both the original financial transaction and its cancellation on the Adjustment page.

Transfer Adjustments

A convenient mechanism exists to transfer moneys between two contracts. The net effect of such a request is the creation of two adjustments, each of which is linked to the other. Both adjustments are created together, frozen together, and posted to the GL together. This is useful because you can't create one side of the transfer and without the other. Refer to How To Create A Transfer Adjustment for more information.

Note: Inter or Intra Account. It's important to be aware that the transfer can be inter or intra account (i.e., the accounts on the two adjustments may be different).

Calculated Adjustments

An adjustment can calculate the adjustment amount using an algorithm. A calculated adjustment can be used to calculate:

- Sales or other taxes for a base amount
- Any charge amount based on a user-supplied quantity, such as feet, miles, hours, etc.
- Others.... Any calculation that can be made by a rate can be applied to an adjustment.

When you create a calculated adjustment, you may be asked to supply a base amount. The user may supply the base amount or the amount could be defaulted from the adjustment type and possibly overridden by the user prior to calculating the adjustment amount.

When the adjustment is generated, the calculation algorithm may use the base amount and calculation date. The base package algorithm calls the rate application and returns the calculated amount with calculation lines and GL distribution codes.

An adjustment's adjustment type controls whether the adjustment is calculated and the algorithm that performs the calculation.

Fastpath: For more information, refer to Adjustment Type Controls Everything and Setting Up Calculated Adjustment Types.

Adjustment Amount May Be Positive, Negative Or Zero

An adjustment's amount may be positive, negative or zero:

- A positive amount causes the customer's balance(s) to increase.
- A negative amount causes the customer's balance(s) to decrease.
- A zero amount will not affect the customer's balance(s). A zero amount is odd, but necessary when you need to use an adjustment to correct the GL distribution. Refer to How To Use An Adjustment To Change Amounts Booked In Your GL for more information.

Copyright © 2024, Oracle and/or its affiliates.

Adjustment Type Controls Everything

When you create an adjustment, you must define its adjustment type. The topics in this section describe how adjustment type controls the behavior of an adjustment.

Controls Which Balance(s) Are Affected

The adjustment type's financial transaction (FT) algorithm controls how payoff balance and current balance are affected by the adjustment amount.

Defines The GL Account Affected By The Adjustment

Most adjustments affect the general ledger (GL) in some way. The following points describe the source of these GL accounts.

- For normal adjustments (i.e., non-calculated adjustments that affect a single contract), there is a single accounting entry generated:
- - One side of the accounting entry is taken from the distribution code on the contract type of the contract affected by the adjustment. For example, if you are adjusting the payoff balance on a normal contract, the A/R account is constructed from the distribution code on the contract's contract type.
  - The other side of the accounting entry is taken from the distribution code on the adjustment's adjustment type. For example, if a late payment charge is created, the "late payment charge" adjustment type references this fee's revenue account.
- For transfer adjustments (i.e., adjustments used to transfer moneys between two contracts), there are two accounting entries generated - one for the "from" side and one for the "to" side. Each adjustment carries its own set of balanced GL accounting details.
- - For each adjustment, one side of the entry is taken from the distribution code on the contract type of the contract affected by the adjustment (just like for normal adjustments).
  - The other sides of both accounting entries have the same GL account. This account should be the intermediate clearing GL account that is to be used for the transfer. The source of this clearing GL account is the distribution code on the adjustment type used to transfer the funds.
- For calculated adjustments (i.e., adjustments where rates may be applied to calculate the adjustment amount), there is a single accounting entry:
- - For each adjustment, one side of the entry is taken from the distribution code on the contract type of the contract affected by the adjustment (just like for normal adjustments).
  - The other side of the entry is taken from the distribution codes that resulted from the rate calculation if the adjustment financial transaction algorithm is set up to use the calculation lines as the distribution code source. If the adjustment financial transaction algorithm is set up to use the adjustment type as the distribution code source, the other side of the accounting entry is taken from the distribution code on the adjustment type.

Note: Not all adjustments affect the GL. As a general rule of thumb, only those adjustments that affect a contract's payoff balance affect the GL. So, if you adjust a budget customer's current balance, but don't adjust their payoff balance, the GL is not affected.

Defaults The Adjustment Amount

The adjustment type may default the adjustment amount in one of the following ways:

- The adjustment type may specify a default amount. This would be used for those adjustment types that have a standard charge for all customers that receive this adjustment, for example a non-sufficient funds charge.
- The adjustment type may specify a default adjustment amount algorithm. This would be used for those adjustment types that have a charge that varies based on other factors. For example, a non-sufficient funds charge may be based on a customer's credit rating.

When an amount is defaulted onto a new adjustment it may be overridden by a user.

Copyright © 2024, Oracle and/or its affiliates.

Calculates The Adjustment Amount

Some adjustment types calculate the adjustment amount. If the adjustment type uses a calculated amount, a base amount is passed to the rate application, which then returns the calculated amount with calculation lines and GL distribution codes.

The adjustment type's generate adjustment algorithm controls which rate is applied to the base amount. A user supplied calculation date controls which version of the rate is used. The user may supply the base amount or it may be defaulted from the adjustment type and possibly overridden by the user prior to calculating the adjustment amount.

Controls The Interface To A/P & 1099 Reporting

If the adjustment type is associated with a payment (e.g. a refund or a loan disbursement) of money to a customer, the adjustment type indicates such with a reference to an A/P request type.

When an adjustment that references an A/P request type is `frozen`, an A/P download request record is created. This record is the interface request to ask your A/P system to cut a check. This interface record is marked with a batch process ID and run number.

- The batch process ID is the process responsible for creating the flat file that contains check request that is interfaced to your account's payable system. The batch process ID is defined on Installation Options - Financial. The base package is supplied with a skeletal background process (referred to by the process ID of `APDL`) that must be populated with logic to format the records in the format compatible with your accounts payable system.
- The run number is the batch process ID's current run number.

Fastpath: Refer to Accounts Payable Check Request for more information.

If the resultant check needs to be reported for income tax purposes under a specific 1099 category, the category is also specified on the adjustment type. The 1099 category is in turn interfaced to the A/P system (the system does NOT manage 1099 reporting).

Controls Information Printed On The Bill

If the adjustment appears on the customer's next bill, the verbiage is specified on the adjustment type (and may NOT be overridden on the adjustment).

Controls If The Adjustment Can Be Frozen Prior To Bill Completion

Refer to Preventing Contract Balances And The GL From Being Impacted Until Bill Completion for how adjustment type can control whether adjustments of a given type can be frozen prior to bill completion.

Controls If The Adjustment Requires Approval

If an adjustment being created online requires approval prior to being frozen, the adjustment type references an appropriate approval profile that defines parameters that control the approval requirements. If the adjustment requires approval, an appropriate approval request is created.

Note: Only online. The system does not create approval requests for adjustments created in batch processing.

Fastpath: For more information about general approval logic, refer to The Big Picture of Approvals.

Unbilled Adjustments And Aged Debt

A `Frozen` adjustment waits in limbo until the customer's next bill is produced. This limbo period could be several weeks if the customer is billed infrequently. When the customer's next bill is completed, all limbo adjustments are linked to the bill. A brief description (as defined on the adjustment type) appears on the printed bill.

It is important to stress the following in respect of these limbo adjustments:

Copyright © 2024, Oracle and/or its affiliates.

- If the adjustment decreases the amount of debt, the customer's aged debt is affected immediately regardless of whether the adjustment appears on a bill.
- If the adjustment increases the amount of debt, the amount the customer owes from an *aged debt* perspective may or may not be affected by the adjustment. There is a switch on the financial transaction created for the adjustment called New Charge that controls the arrears behavior. When the FT for this type of adjustment is created, the switch is set to on and the customer's aged debt will not reflect the adjustment amount until the adjustment is swept onto a bill. The moment the adjustment is swept onto the customer's bill, the debt starts aging. A user may turn the switch off to cause the debt to start aging sooner. For this case, the date on which the adjustment starts aging must be defined in the Arrears Date field.
- The amount a customer owes *in total* is immediately affected by the adjustment regardless of whether the adjustment appears on a bill. This means that amount of aged debt may not be in sync with the total amount owed. This seems odd but is useful from a credit and collections perspective. You see, you probably don't want to start aging an adjustment until the customer has actually seen it.

An Adjustment May Affect More Than Just Customer Balances

Fastpath: When an adjustment is frozen or cancelled, a customer's current and payoff balances are affected. However, several other objects may be affected when such events occur. Refer to Obscure Things That Can Happen for more information.

Maintaining Adjustments

An adjustment is used to change the amount of debt stored on a contract. The topics in this section describe the pages on which adjustments are maintained.

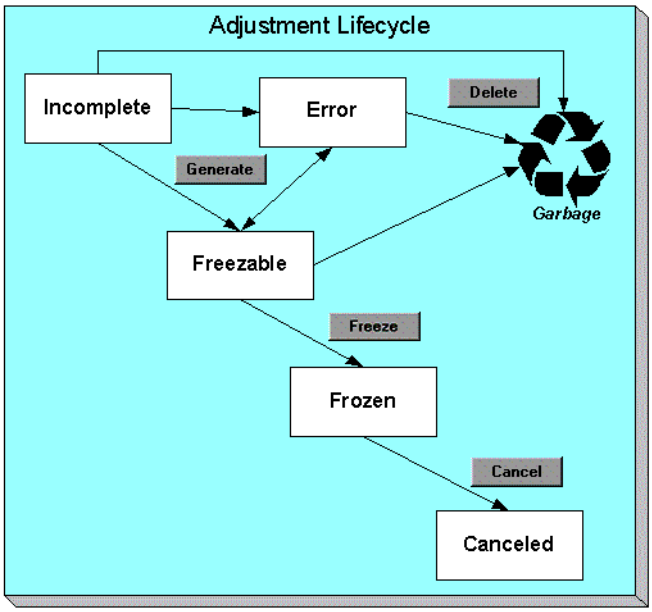
Fastpath: For more information about adjustments, refer to The Big Picture Of Adjustments.

The Lifecycle Of An Adjustment

The following diagram shows the possible lifecycle of an adjustment.

CAUTION: This diagram only makes sense in the context of the page used to maintain Adjustments. Refer to Adjustments - Main Information for the details.

Copyright © 2024, Oracle and/or its affiliates.



Adjustments are initially created in the `Incomplete` state. Adjustments in this state don't have a financial transaction. This means, you can change the adjustment type and amount at will.

Click **Generate** to generate a financial transaction for the adjustment. The financial transaction contains the adjustment's effect on the general ledger and on the customer's payoff and current balances. If the adjustment is calculated, the algorithm on the adjustment type's Generate Adjustment event controls how the adjustment is calculated. If the adjustment is calculated, you must specify the calculation date, which is passed as a parameter to the calculation algorithm and is used for calculations that are effective dated (e.g., rate version or bill factor value). The financial transaction (FT) algorithm defined on the adjustment type Adj. Financial Transaction event controls how the financial transaction is constructed. For calculated adjustments, the distribution code may be taken from the adjustment type or the calculation lines.

After generating the financial transaction, the adjustment becomes `Freezable`. While in this state, you may change the contract, adjustment type and amount at will. However, if you change this information, you will have to regenerate the financial transaction (by clicking the Generate button).

In the very rare situation when the system cannot generate the financial transaction because of inconsistent setup data, the adjustment is moved to the `Error` state. You may regenerate the financial transaction after correcting the source of the error. You may also delete such an adjustment.

Click the delete button to physically remove an `Incomplete`, `Error` or `Freezable` adjustment from the database.

CAUTION: Freeze may not be enabled. Please be aware that you can prevent specific types of adjustments from being frozen until the account's next bill is `completed`. Refer to Preventing Contract Balances And The GL From Being Impacted Until Bill Completion for how to enable this feature on your adjustment types.

Click **Freeze** to freeze the adjustment and its financial transaction. After doing this, the adjustment's state becomes `Frozen`. While in this state, you cannot change the adjustment's type or amount, but you may change:

- When the adjustment starts aging.
- Whether the adjustment appears on a customer's bill.
- The accounting date used to derive the general ledger accounting period(s) to which the financial transaction is booked.

Note: Adjustments may require approval. You cannot manually freeze adjustments whose adjustment type refers to an approval profile. If an adjustment of this type does not already have an approval request, a `Submit` button replaces the **Freeze** to enable request for an approval. For `freezable` adjustments that have an active approval request, all action buttons on the adjustment page are disabled.

Copyright © 2024, Oracle and/or its affiliates.

If you need to remove the financial effects of an adjustment, click **Cancel**. Canceling an adjustment causes the generation of another financial transaction. This new financial transaction reverses the financial impact of the original adjustment.

Adjustments - Main Information

Fastpath: The **Description of Page** section below describes the fields on this page. Refer to How and When To Use An Adjustment for instructions describing how to perform common maintenance functions.

The Main page contains core adjustment information. Open this page using **Billing Management, Adjustment**.

CAUTION: All adjustments reference a very important field that controls much validation and processing; this field is called Adjustment Type. Take special care when adding a new adjustment to specify the appropriate Adjustment Type as it affects how the adjustment appears on the customer's bill, how the adjustment is reflected in your general ledger, and much more. After the adjustment is frozen, you may not change its Adjustment Type.

Description of Page

Adjustment Info contains a concatenation of the adjustment amount, adjustment type and status.

Note: Formatting may be performed by a plug-in. The contents of **Adjustment Info** may be formatted by a plug-in algorithm on the Adjustment Type. Refer to the base package's C1-ADT-INFO for an example. If such an algorithm is not plugged-in on the Adjustment Type, the system looks for a corresponding algorithm on the installation record. Refer to the base package's C1-ADI-INFO for examples. If you prefer different formatting logic, your system administrator should configure the system appropriately.

Adjustment ID is the system-assigned, unique identifier of the adjustment.

Account ID is the account to which the adjustment is linked. The name of the main person on the account appears next to the account ID.

Use **Contract** ID to define the contract whose value needs to be adjusted. Basic information about the contract appears adjacent.

Indicate the **Adjustment Type**. This field is very important as it controls numerous aspects of the adjustment's impact on the customer's balance and your general ledger. This field is gray after the adjustment is frozen.

Note: You can only choose certain adjustment types. The contract's contract type has a collection of valid adjustment profiles. You may only reference adjustment types that are listed in one of the adjustment type profiles linked to the contract type.

Select the **currency** when you want to create an adjustment in a currency which is different from the account's invoice currency. This field is disabled when either of the following conditions is true:

- The **Allow Multi-Currency Accounts** option type of the **C1_MLTCURACC** feature configuration is set to **N**.
- The account's customer class is disabled for open item accounting. In other words, the **Open Item Accounting** check box is not selected for the account's customer class.

For more information on how these adjustments are swept onto the bill, refer to Generating Bills.

Enter the **Amount** of the adjustment. If the adjustment type is a calculated adjustment, the **Calculated Amount** and **Calculation Date** are displayed. The values of the calculated amount and calculation date are displayed after the adjustment is generated. The calculated amount shows the result of the generate adjustment algorithm. The calculation date is specified when you click Generate for a calculated adjustment type. It is used by the generate adjustment algorithm for any calculations that are effective dated (e.g., rate version or bill factor value).

Note: The adjustment amount defaults based on the adjustment type. When you change the adjustment type, the amount changes accordingly. You may change the adjustment amount after it is defaulted.

Use **Bill ID** to indicate the previous completed bill against which you want to create an adjustment. The **Bill ID** field appears in the **Adjustment** screen when the **Print by Default** check box is selected on the adjustment type.

Copyright © 2024, Oracle and/or its affiliates.

The **Original Amount and Currency** field displays the adjustment amount and currency before currency conversion. The **Exchange Rate** field displays the exchange rate value used during the currency conversion. The **Exchange Rate ID** field displays the ID that uniquely identifies the exchange rate used during the currency conversion. These three fields are displayed when the currency conversion takes place on the adjustment.

On Behalf Contract ID is a display-only field that is only populated on system-generated adjustments instigated by one customer, but attributed to another customer. This field is only visible if it is populated on the adjustment.

You may not change an adjustment's **Adjustment Status** directly. Rather, you use the buttons in the **Adjustment Actions** area. Refer to The Lifecycle Of An Adjustment for the details.

If the status is `canceled`, the **Cancel Reason** is displayed.

Use the **Comments** to describe anything unusual about the adjustment.

The **Creation Date** defines the date on which the adjustment was created.

If a related approval request exists for this adjustment, a message displays indicating that there is an approval request.

The financial transaction (FT) grid contains the financial transactions associated with the adjustment. It only contains information after the adjustment is frozen. If the adjustment is canceled, a second row appears showing the details of the cancellation FT.

- **Financial Transaction ID** is the system-assigned unique identifier of the FT. Click the go to button to transfer to the financial transaction. On this page you can change certain aspects of the FT in question.
- **Arrears Date** is the date the FT starts aging.
- **Accounting Date** is the date the system uses to determine the FT's accounting period in your general ledger.
- **Current Amount** contains the FT's effect on the contract's current balance.
- **Payoff Amount** contains the FT's effect on the contract's payoff balance.
- **Bill ID** is the bill on which the FT appears (if it has been swept onto a bill). Click the adjacent go to button to transfer to the bill on which the FT appears. Note: an FT is linked to a bill the next time a bill is completed for the contract's account.

The **Calculation Lines** grid contains the details of the calculations associated with a calculated adjustment. It only appears if the rate that calculated the adjustment amount created at least one calculation line. One row exists for every calculation involved in the process. This information is for audit purposes only and cannot be modified. The following information displays in the grid:

- If at least one of the calculation lines has characteristics, **Calc Line Char** displays go to buttons, allowing you to go to characteristics that are linked to a specific adjustment calculation line. This column is only displayed if at least one of the calculation lines has characteristics.
- **Sequence** is the system-assigned unique identifier of the calculation detail row.
- **Description on Bill** is the information about the calculation line that appears on the customer's bill.
- **Calculated Amount** is the calculated amount associated with the calculation line.
- The **Print** switch controls whether information about this line will print on the customer's bill.
- The **Appears in Summary** switch defines if this line's amount also appears on a summary line. This switch plays a part at bill print time - those lines that appear in a summary print in the left dollar column, those that don't appear in a summary print in the right dollar column.
- Bill segments created by applying a rate have this switch turned on if the corresponding rate component is summarized on a summary rate component.
- **Unit of Measure** (UOM) is the unit of measure of the service quantity priced on the calculation line.
- **Sub-Unit of Measure** (Sub-UOM) is the sub-unit of measure code of the service quantity priced on the calculation line.
- **SQI** is the service quantity identifier of the service quantity priced on the calculation line.
- **Billable Service Quantity** is the service quantity priced on the calculation line. This quantity differs from the product consumption if there are SQ rules or register rules in effect.
- **Base Amount** is used by calculation lines (e.g. taxes) that are cross-referenced to other calculation lines and whose value(s), therefore, depend on the amounts calculated by those other lines. The Base Amount shows the total amount derived from the cross-referenced line(s) that the current line then used to calculate its billed amount.

Copyright © 2024, Oracle and/or its affiliates.

- **Rate Comp** onent **Seq** uencerefers to the sequence number of the rate component on the applicable rate version that was used to calculate the line.
- **Measures Peak Qty** is checked if the UOM priced on the calculation line is used to measure a peak quantity.
- **Exempt Amount** is the amount of the calculated charge that the customer doesn't have to pay because they are tax exempt.
- **Distribution Code** is the distribution code associated with the calculation line. This distribution code is used to build the general ledger details on the bill segment's financial transaction.
- **Description** describes the characteristic value that was used when the line's amount was calculated. This information is only displayed if the line was calculated using a bill factor (because only bill factors use characteristic values). Refer to *An Illustration Of A Bill Factor And Its Characteristics* for more information.

Fastpath: For more information, refer to *Calculated Adjustments*.

The **Adjustment Actions** area contains buttons that you use to commit and cancel the adjustment's financial impact. The adjustment's status controls the button you can see and select. Refer to The Lifecycle Of An Adjustment for the details.

CAUTION: Freeze may not be enabled. Please be aware that you can prevent specific types of adjustments from being frozen until the account's next bill is `completed`. Refer to Preventing Contract Balances And The GL From Being Impacted Until Bill Completion for how to enable this feature on your adjustment types.

CAUTION: Cancel may not be enabled. The cancel button is not enabled if the adjustment is linked to a bill that is written off.

Note: Canceling An Accounts Payable (A/P) Adjustment. If you need to cancel an A/P adjustment (i.e. an adjustment of an adjustment type that has an A/P request type) that has already been extracted by A/P, refer to How To Cancel An A/P Adjustment After It's Been Selected By A/P.

Adjustments - Characteristics

You use this page to link additional information to the adjustment. Open using **Billing Management, Adjustment, Characteristics**.

Description of Page

The characteristics collection contains information that describes miscellaneous information about the adjustment.

Note: Characteristic Types. You can only choose characteristic types defined on the adjustment's adjustment type.

The following fields display:

Characteristic Type Indicate the type of characteristic.

Characteristic Value Indicate the value of the characteristic.

Note: An adjustment's characteristics default from the adjustment type.

Adjustments - Transfer Adjustment

Fastpath: The **Description of Page** section below describes the fields on this page. Refer to How and When To Use An Adjustment for instructions describing how to perform common maintenance functions.

The Transfer Adjustment is used to define the reciprocal adjustment associated with a transfer adjustment (there are always two adjustments associated with a transfer). Open using **Billing Management, Adjustment, Transfer Adjustment**.

Description of Page

Copyright © 2024, Oracle and/or its affiliates.

Adjustment Info contains a concatenation of the adjustment amount, adjustment type and status.

Adjustment ID is the system-assigned, unique identifier of the adjustment.

Account ID is the account to which the adjustment is linked. The name of the main person on the account appears next to the account ID.

Use **Contract** ID to define the transfer to contract. Basic information about the contract appears adjacent.

The **Adjustment Type** and adjustment **Amount** from the first page are displayed. If you need to change either value, return to the first page.

Adjustment Status shows the status of the transfer to adjustment. If you need to change the adjustment's status, use the action buttons.

Use the **Comments** to describe anything unusual about the adjustment.

Transfer Adj ID is the system-assigned, unique identifier of the adjustment.

The **Creation Date** defines the date on which the adjustment was created.

The financial transaction grid contains the financial transactions associated with the adjustment. It only contains information after the adjustment is frozen. If the adjustment is canceled, a second row appears showing the details of the cancellation FT.

- **Financial Transaction ID** is the system-assigned unique identifier of the FT. Click the go to button to transfer to the financial transaction. On this page you can change certain aspects of the FT in question.
- **Arrears Date** is the date the FT starts aging.
- **Accounting Date** is the date the system uses to determine the FT's accounting period in your general ledger.
- **Current Amount** contains the FT's effect on the contract's current balance.
- **Payoff Amount** contains the FT's effect on the contract's payoff balance.
- **Bill ID** is the bill on which the FT appears (if it has been swept onto a bill). Click the adjacent go to button to transfer to the bill on which the FT appears. Note: an FT is linked to a bill the next time a bill is completed for the account.

Adjustments - A/P Request

Fastpath: The **Description of Page** below describes the fields on this page. Refer to How and When To Use An Adjustment for instructions describing how to perform common maintenance functions.

The A/P Request page contains information about adjustments used to refund money via an A/P check request. This page is only relevant if:

- The adjustment's adjustment type references an A/P Request Type (i.e., it will be interfaced to your accounts payable system which will cut the check), AND
- The adjustment is `frozen`.

Open this page using **Billing Management, Adjustment, A/P Request**.

Description of Page

Adjustment Info contains a concatenation of the adjustment amount, adjustment type and status.

Adjustment ID is the system-assigned, unique identifier of the adjustment.

Account ID is the account to which the adjustment is linked.

Note: Adjustment type controls A/P check requests. The A/P Request information is populated when an adjustment used to refund money via an A/P check request is `frozen`. Whether or not an adjustment is interfaced to A/P is controlled by the adjustment's adjustment type. The adjustment type's A/P Request Type controls the payment date and the bank.

Name is the name printed on the check. This name is derived from the account's main person:

Copyright © 2024, Oracle and/or its affiliates.

- If this person has an override mailing name, the first line of the override mailing name is used.
- If this person does not have an override mailing name, the person's primary name is used.

Payment Selection Status is the status of the check request. The values are:

- `Not Selected for Payment` before it's selected by A/P for payment
- `Paid` after it's selected by A/P
- `Canceled` if it's been canceled in A/P
- `Hold if it's been held in A/P`

Scheduled to Pay is the date on which the check is scheduled to be cut. This is equal to the adjustment date plus the Due Days on the adjustment type's A/P request type.

The following fields are provided in order to support a two-way interface with an A/P system. The delivered system does not have processes which update these fields.

- **Payment Date** is the date on which the check was cut in A/P. This field is only populated after A/P cuts the check.
- **Paid Amount** is the amount of the check. This field is only populated after A/P cuts the check.
- **A/P Request ID** is the system-assigned, unique identifier of the A/P check request.
- **Payment Number** is the system-assigned number of the payment in A/P (this number typically appears on the printed check). This field is only populated after A/P cuts the check.

The address information in the bottom frame is the address to which the check is mailed. This is the billing address of the main customer linked to the account. After the adjustment is frozen, this information is automatically populated based on the address information for the account. This information is modifiable until the status is `Paid`.

Adjustments - Approval

This page contains zones that show information about an approval request, and it is available when the adjustment being displayed references an approval request.

Open this page using **Billing Management, Adjustment, Approval**.

Description of page

This page includes the following zones:

- A zone that displays standard Adjustment Information.
- A zone that displays information relevant for **Approval Request**
- A standard log zone displays the approval request log.

Note: On rejecting an adjustment, a To Do entry is created using the **C1-ADJRJ** To Do type. You can search for the To Do entry in the **To Do List** screen. On clicking the **Submitted Message** link corresponding to the To Do entry, the **Adjustment** screen appears with the details of the account whose adjustment was rejected. You can then create a new adjustment for the account.

Fastpath: For more information, refer to The Big Picture of Approvals.

Financial - Adjustment Calculation Line Characteristics

This page displays the characteristics that are linked to a specific adjustment calculation line. The information on this page is for audit purposes only and cannot be changed.

Open this page by clicking on the characteristics go to button on the calculation lines grid of the Adjustments - Main page.

Description of Page

Copyright © 2024, Oracle and/or its affiliates.

Account is the account to which the adjustment is linked. The name of the main person on the account appears next to the account ID.

The **Contract** is the contract whose value needs to be adjusted. Basic information about the contract appears adjacent.

The **Adjustment Info** displays information to identify the adjustment, including amount, type, and status.

Adjustment ID displays the identification of the adjustment, and **Calc Line** displays the calculation line with which the characteristics are associated.

The **Description on Bill** displays the description of the calculation line that appears on the customer's bill.

The characteristics grid displays the **Characteristic Type** s and the **Characteristic Value** s associated with the calculation line.

Fastpath: For more information about characteristics, refer to Setting Up Characteristic Types and Their Values.

How and When To Use An Adjustment

Note: A last resort. Adjustments should be viewed as a last resort because you should correct the cause of errors, rather than the effect. For example, if a customer's bill is incorrect, you should fix the customer's rate / contract terms and then perform a cancel / rebill. Issuing an adjustment to fudge the balance is really just that - a fudge.

The topics in this section describe how to perform common adjustment maintenance functions.

How To Create A Transfer Adjustment

The following steps describe how to create a transfer adjustment on the Adjustment page.

Note: You cannot use an adjustment type with a calculated amount for transfer adjustments.

CAUTION: These steps only make sense in the context of the page used to maintain Adjustments. Refer to Adjustments - Main Information for the details.

To create a transfer adjustment:

- Specify the "transfer from" account and contract on the Adjustment - Main page.
- Specify the adjustment type and the amount and transfer to the Transfer Adjustment page.
- Specify the "transfer to" contract and click the **Generate** and **Freeze** buttons to generate and freeze the adjustments' financial transactions. Please be aware that the **Freeze** button may be disabled for certain types of adjustments. This is because you can prevent specific types of adjustments from being frozen until the account's next bill is `completed`. Refer to Preventing Contract Balances And The GL From Being Impacted Until Bill Completion for how to enable this feature on your adjustment types.

How To Create A Calculated Adjustment

The following steps describe how to create a calculated adjustment on the Adjustment page.

CAUTION: These steps only make sense in the context of the page used to maintain Adjustments. Refer to Adjustments - Main Information for the details.

To create a calculated adjustment:

1. Open **Billing Management, Adjustment, Main** and specify the **Contract** to which the adjustment will be applied.
2. Specify the **Adjustment Type** that calculates the amount, and enter a base **Amount**.

Copyright © 2024, Oracle and/or its affiliates.

3. Click **Generate**. The **Generate** dialog box opens.
4. In the resulting dialog box, specify an **Accounting Date** and **Generation Reference Date**. Click **Calculate**.
5. Click **Freeze** to freeze the calculated transactions.

The **Calculation Date** that you supply is used by the generate adjustment algorithm for any calculations that are effective dated (e.g., rate version or bill factor value).

Please be aware that the **Freeze** button may be disabled for certain types of adjustments. This is because you can prevent specific types of adjustments from being frozen until the account's next bill is `completed`. Refer to Preventing Contract Balances And The GL From Being Impacted Until Bill Completion for how to enable this feature on your adjustment types.

How To Cancel An A/P Adjustment After It Has Been Selected By A/P

Adjustments that are interfaced to A/P (because A/P needs to cut a refund check) sometimes need to be canceled. You may cancel an A/P adjustment in the CIS system while its A/P request is `Not Selected`. If the A/P request is `Paid` you must first cancel the payment in A/P. Canceling the payment in A/P changes the A/P request in CIS to `Canceled`. At this point, you can cancel the adjustment in CIS.

How To Correct Normal Contract Debt

If a customer is unhappy with the balance on a "normal" contract and you cannot fix the source of their unhappiness (i.e., the bills and payments), create an adjustment. Simply select an adjustment type that describes why you are adjusting the balance and enter the amount of the adjustment (a negative number if you are crediting the customer).

Note: What's normal? There really is no such thing as a "normal" contract. We use this term to reference contracts used to bill for services like financial service, land leases, home warranties, security monitoring, etc. where the current amount is the same as the payoff amount.

CAUTION: A contract that is part of a budget plan is NOT normal. Refer to How To Adjust A Contract On A Budget **for more information**.

How To Apply Adhoc Fees To A Contract

If you need to apply a fee (e.g., late payment charge, connection charge, NSF charge) to a "normal" contract, issue an adjustment using the appropriate adjustment type. The adjustment amount should be a positive number so that the customer's balance increases.

CAUTION: Take care if you apply a fee to a contract that's on a budget. The adjustment affects current balance and payoff balance by the amount of the fee. If you didn't mean to affect the customer's current balance, you'll have to issue another adjustment to change it.

How To Apply Interest To A Deposit

Fastpath: You should only have to issue adjustments to calculate and refund interest if you need to override the system's automated procedures.

If you need to manually apply interest to a deposit, issue an adjustment using the appropriate adjustment type. The adjustment amount should be a negative number (you are crediting the customer's deposit balance).

If you want to apply the interest to the customer's outstanding balance, refer to How To Apply A Deposit To Outstanding Debt .

Copyright © 2024, Oracle and/or its affiliates.

If you want to cut a check for the deposit, refer to How To Refund A Deposit With A Check .

How To Apply A Deposit To Outstanding Debt

Fastpath: You should only have to issue adjustments to apply a deposit to outstanding debt if you need to override the system's automated procedures.

Applying a deposit is a two-step process:

- The deposit contract's payoff balance contains the amount of deposit you are holding on behalf of the customer's debt. Its current balance should be zero (i.e., the customer doesn't owe any additional deposit). Before you apply the deposit balance, we recommend making the deposit's current balance equal to the amount to be transferred. You do this by issuing an adjustment that just affects the deposit's current balance. The amount of this adjustment should be a negative number.
- After the deposit's current balance reflects an appropriate credit amount (equal to the amount to be applied to the customer's debt), use the Transfer Adjustment process to apply the deposit's credit balance to another contract's debit balance. The deposit contract is the transfer from contract; the delinquent contract is the transfer to contract. The adjustment amount should be a positive number. Refer to How To Create A Transfer Adjustment for more information.

CAUTION: Take care if you apply a deposit to a contract that's on a budget. The transfer adjustment affects current balance and payoff balance by the same amount. If you didn't mean to affect the customer's current balance, you'll have to issue another adjustment to change it.

How To Refund A Deposit With A Check

Fastpath: You should only have to issue adjustments to apply a deposit to outstanding debt if you need to override the system's automated procedures.

Refunding a deposit with a check (cut by your A/P system) is a two step process:

- The deposit contract's payoff balance contains the amount of deposit you are holding on behalf of the customer's debt. Its current balance should be zero (i.e., the customer doesn't owe any additional deposit). Before you refund the deposit balance, we recommend making the deposit's current balance equal to the amount to be refunded. You do this by issuing an adjustment that just affects the deposit's current balance. The amount of this adjustment should be a negative number.
- After the deposit's current balance reflects an appropriate credit amount (equal to the amount to be refunded), issue a refund adjustment. The adjustment amount should be a positive number. This adjustment causes a check request to be interfaced to your A/P system.

How To Write-Down Deposit Due

If a customer doesn't pay their deposit and you don't expect them to, create an adjustment to write-down the current balance. Simply select an adjustment type that describes why you are adjusting the balance and enter the amount of the adjustment (a negative number).

How To Refund A Credit Balance With A Check

If you need to refund a credit balance on a "normal" contract, issue a refund adjustment using the appropriate adjustment type. The adjustment amount should be a positive number. This adjustment causes a check request to be interfaced to your A/P system.

Copyright © 2024, Oracle and/or its affiliates.

CAUTION: Take care if you refund a credit balance from a contract that's on a budget. The adjustment affects current balance and payoff balance by the amount of the refund. If you didn't mean to affect the customer's current balance, you'll have to issue another adjustment to change it.

How To Transfer A Credit Balance To A Contract In Arrears

Use the Transfer Adjustment process to apply a "normal" contract's credit balance to another contract's debit balance. The credit contract is the transfer from contract; the delinquent contract is the transfer to contract. The adjustment amount should be a positive number. Refer to How To Create A Transfer Adjustment for more information.

CAUTION: Take care if you apply a credit to a contract that's on a budget. The transfer adjustment affects current balance and payoff balance by the transfer amount. If you didn't mean to affect the customer's current balance, you'll have to issue another adjustment to change it.

How To Change The Age Of Debt

If a contract's debt is older (or younger) than what it should be (for whatever reason), you should do the following.

- Use the Transfer Adjustment process to re-age the debt. The transfer from and transfer to contracts will be the same. The adjustment amount should be a negative number. Refer to How To Create A Transfer Adjustment for more information.
- After creating the transfer adjustment, drill into the transfer to adjustment and populate the arrears date with the day on which the debt should start aging (e.g., if the debt should be 7 days old, the arrears date should be the current date minus 7 days).

CAUTION: Be careful if you transfer debt from a contract that's on a budget. Before doing the transfer, you should make the customer's current balance equal their payoff balance.

How To Write-Down A Charitable Contribution Contract Debt

If a customer doesn't want to contribute as much to the charity as the agreed to, create an adjustment to fix the current balance (charity contracts shouldn't have a payoff balance as the "debt" is voluntary). Simply select an adjustment type that describes why you are adjusting the balance and enter the amount of the adjustment (a negative number).

How To Write-Off Debt

To write-off debt:

CAUTION: You don't have to do the tasks outlined below if you use the Write Off - Main page. This page does these tasks for you.

- Create a write-off contract using the Start Service page (you could also use the contract page to do this).
- Use the Transfer Adjustment process to transfer the delinquent debt to the to the write-off contract. The original contract is the transfer from contract; the write-off contract is the transfer to contract. The adjustment amount should be a negative number. Refer to Transfer Adjustments for more information.

CAUTION: Take care if you transfer debt from a contract that's on a budget. Before doing the transfer, you should make the contract's current balance equal their payoff balance.

Copyright © 2024, Oracle and/or its affiliates.

How To Cancel A Contract's Budget

Canceling a contract's budget involves the following steps:

- Update the contract to indicate the contract is no longer on a budget.
- Issue an adjustment to make the contract's current balance equal the payoff balance.

> **Note: Credit and collections (C&C).** You can change the arrears date on the adjustment that synchronizes current and payoff balance to reflect the true age of the debt. You do this on the financial transaction associated with the adjustment. If you do this, C&C will monitor the old debt accordingly.

How To Adjust A Contract On A Budget

If you need to adjust the debt on a contract that's on a budget plan, you have to ask yourself why you're adjusting the balance:

- If you're adjusting the balance to correct the payoff balance (for whatever reason), choose an adjustment type that just affects this balance.
- If you're adjusting the balance to correct the current balance (for whatever reason), choose an adjustment type that just affects this balance.
- If you need to adjust both balances equally, choose an adjustment type that affects both.

How To Adjust An Installment Plan Contract

If you need to adjust the debt on a contract that's used to bill a total amount in installments, you have to ask yourself why you're adjusting the balance:

- If you're adjusting the balance to correct the payoff balance (for whatever reason), choose an adjustment type that just affects this balance.
- If you're adjusting the balance to correct the current balance (for whatever reason), choose an adjustment type that just affects this balance.

If you need to adjust both balances equally, choose an adjustment type that affects both.

How To Use An Adjustment To Change The GL Distribution

Assume you have a frozen bill segment that affected an incorrect GL distribution code (e.g., it booked revenue to Banking - revenue and it should have booked it to Banking Tax Liability - revenue. You have two ways to correct such a situation:

- You could correct the cause of the incorrect distribution code (probably incorrect GL information on a rate component) and cancel / rebill the bill segment. Refer to Cancel / Rebill for more information about this method.
- You could create an adjustment that does not impact the customer's balance but does have a GL entry. This GL entry would reverse the effect of the original distribution code and have an offsetting entry to the correct distribution code. The following points explain how to do this:
-
  - Choose an adjustment type that only impacts the GL (i.e., the adjustment's adjustment type has an FT algorithm that doesn't impact the customer's current or payoff balance, it only impacts the GL).
  - Enter an adjustment amount of zero.
  - Generate the adjustment.
  - Drill into the adjustment's financial transaction and enter two (or more) GL distribution rows that reflect the redistribution of the amounts.
  - Return to the adjustment and freeze it.

Copyright © 2024, Oracle and/or its affiliates.

Note: Adjusting statistical amounts. You can use the above approach to change the statistical amount that is interfaced to the general ledger. The statistical amount is simply another column on the financial transaction's GL distribution rows.

How to Create an Adjustment against a Completed Bill

The system allows you to create an adjustment against the previous completed bills of an account. When you create an adjustment against the previous completed bill, the previous bill balance is affected, but the adjustment is presented on the next bill of the account. When the next bill is completed, the presentment bill ID is stamped on the frozen adjustment FT.

You can create adjustments against the previous bill when the **Print By Default** check box is selected on the adjustment type. While creating and freezing an adjustment which is created against the previous bill, the system stamps the match event ID of the previous bill on the adjustment FT. This happens when you create an algorithm of the **C1-GTPRBLME** algorithm type and attach it to the **Adjustment Freeze** algorithm spot on the adjustment type. This algorithm checks whether there is any unbalanced match event on the previous bill. If the unbalanced match event exists on the previous bill, the match event ID is stamped on the adjustment FT. However, if the unbalanced match event does not exist on the previous bill, the match event ID is not stamped on the adjustment FT.

At times, as a result of credit adjustments created against a previous completed bill, the previous completed bill balance may change from debit to credit. If the previous bill amount is in credit, the system must transfer the credit on the overpayment contract of the account by creating the transfer adjustment. This happens when you create an algorithm of the **C1-BLCRXFR** algorithm type and attach it to the **Adjustment Post Freeze** algorithm spot on the adjustment type.

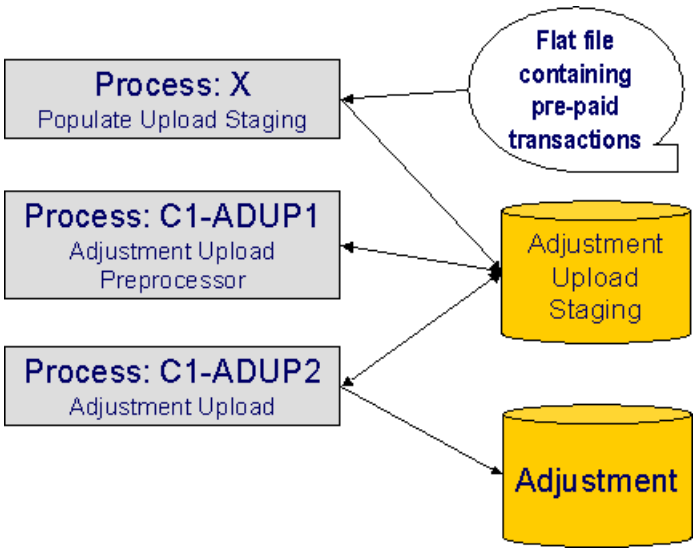
If the overpayment contract is not available on the account, the system first creates the overpayment contract on the account and then creates the transfer adjustment. If the match event ID is available on the adjustment FT, the system stamps the match event ID on the bill side of the transfer adjustment. However, if the match event ID is not available on the adjustment FT, the system creates the match event and stamps the match event ID on all FTs of the previous bill and on the bill side of the transfer adjustment.

Interfacing Adjustments From External Sources

The topics in this section describe how adjustments are uploaded from an external source.

Interfacing Adjustments

The following diagram illustrates the processes involved in the uploading of adjustments into the system.

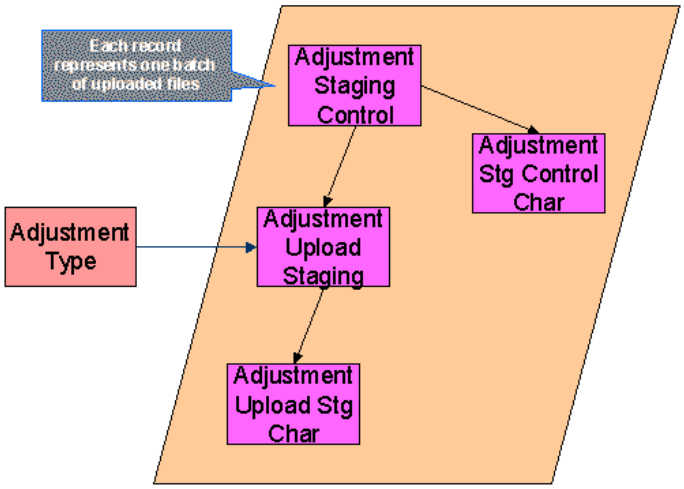


Copyright © 2024, Oracle and/or its affiliates.

The topics in this section describe how these processes work.

Process X - Populate Adjustment Upload Records

Process X refers to the mechanism used by your organization to populate the various staging tables (shown in the orange section of the following ERD).



The topics in this section describe each of these tables.

Adjustment Staging Control

You must create an adjustment staging control record for each batch of adjustments to be uploaded into the system. The name of this table is **CI_ADJ_STG_CTL**.

The following table lists and describes each column of the **CI_ADJ_STG_CTL** table:

Column Name	Length	Req'd	Data Type	Comments
ADJ_STG_CTL_ID	12	Y	N	This is the unique identifier of the adjustment staging control record. This key is a sequential number so you can use a database function to assign the value when populating the table.
CRE_DTTM	15	Y	DateTime	This is the date / time on which the adjustment staging control record was created. This must be populated with the current date / time.
ADJ_STG_CTL_STATUS_FLG	4	Y	A/N	This must be set to `P` for `Pending`.
ADJ_STG_UP_REC_CNT	10	Y	N	This is the total number of adjustment upload staging records that are linked to this adjustment staging control.
TOT_ADJ_AMT	13.2	Y	N	This column must equal the sum of adjustment amounts on the adjustment upload staging records that are linked to this adjustment staging control.

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Length	Req'd	Data Type	Comments
CURRENCY_CD	3	Y	A/N	This must be a valid currency code in the system. Refer to Defining Currency Codes for more information.
MESSAGE_CATEGORY	5	N	N	Leave this blank. The adjustment upload preprocessor populates this when an error occurs during upload.
MESSAGE_NBR	5	N	N	Leave this blank. The adjustment upload preprocessor populates this when an error occurs during upload.

Adjustment Staging Control Characteristic

You must create an adjustment staging control characteristic record for each characteristic that you would like to link to the adjustment staging control. The name of this table is **CI_ADJ_STG_CTL_CHAR**.

The following table lists and describes each column of the **CI_ADJ_STG_CTL_CHAR** table:

Column Name	Length	Req'd	Data Type	Comments
ADJ_STG_CTL_ID	12	Y	N	This must correspond with the prime key of the related CI_ADJ_STG_CTL record.
CHAR_TYPE_CD	8	Y	A/N	This must correspond with a characteristic type that is defined as valid for `adjustment staging control`. Refer to Setting Up Characteristic Types & Their Values for more information.
SEQ_NUM	3	Y	N	This should be set to `10` unless you have multiple values for a given adjustment staging control and characteristic type.
CHAR_VAL	16	N	A/N	Populate this field if your characteristic type is `predefined`.
ADHOC_CHAR_VAL	254	N	A/N	Populate this field if your characteristic type is `ad-hoc` or `file location`.
CHAR_VAL_FK1 - CHAR_VAL_FK5	50 each	N	A/N	Populate these fields if your characteristic type is `foreign key reference`. Up to five columns of 50 bytes each are provided to accommodate compound keys.

Adjustment Upload Staging

You must create an adjustment upload staging record for each adjustment you want to upload. The name of this table is **CI_ADJ_STG_UP**.

The following table lists and describes each column of the **CI_ADJ_STG_UP** table:

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Length	Req'd	Data Type	Comments
ADJ_STG_UP_ID	12	Y	N	This is the unique identifier of the adjustment upload staging record. This key is a sequential number so you can use a database function to assign the value when populating the table.
ADJ_STG_CTL_ID	12	Y	N	The ID of the adjustment staging control that is linked to this adjustment upload staging record.
ADJ_TYPE_CD	8	Y	A/N	This must correspond to the prime key of one of your adjustment types. Refer to Setting up Adjustment Types for more information.
ADJ_STG_UP_STATUS_FLG	4	Y	A/N	This must be set to `P` for `Pending`.
CREATE_DT	10	Y	Date	The date when the adjustment occurred.
ADJ_AMT	13.2	Y	N	The amount of the adjustment.
ADJ_SUSPENSE_FLG	4	Y	N	This must be set to `NSUS` for `Not In Suspense`.
SA_ID	10	N	A/N	This must correspond to a valid contract in the system. If you leave this blank and opt to let the system find the contract during adjustment upload preprocessing, a `Determine Contract` algorithm must be plugged in on the associated adjustment type. This plug-in is meant to derive the contract ID based on supplied miscellaneous information - e.g. information supplied via adjustment characteristic upload staging.
ADJ_ID	10	N	A/N	Leave this blank. The adjustment upload process populates this.
SUSPENSE_ADJ_ID	10	N	A/N	Leave this blank. The adjustment upload process populates this.
MESSAGE_CATEGORY	5	N	N	Leave this blank. The adjustment upload background processes populate this when an error occurs during upload.
MESSAGE_NBR	5	N	N	Leave this blank. The adjustment upload background processes populate this when an error occurs during upload.

Copyright © 2024, Oracle and/or its affiliates.

Adjustment Characteristic Upload Staging

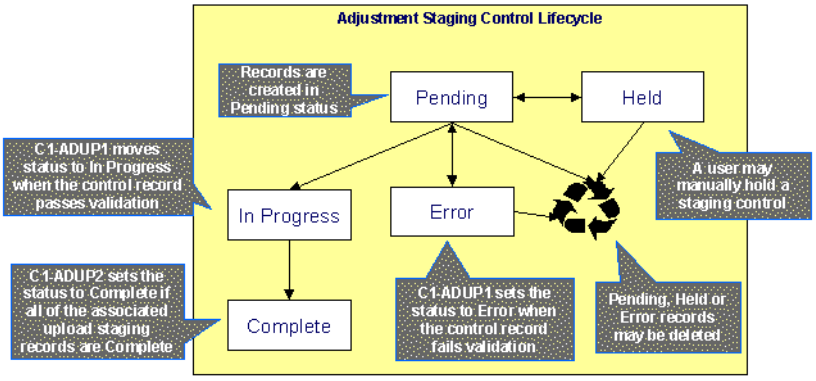
You must create an adjustment characteristic upload staging record for each characteristic that you would like to link to the adjustment upload staging. The name of this table is **CI_ADJ_STG_UP_CHAR**.

The following table lists and describes each column of the **CI_ADJ_STG_UP_CHAR** table:

Column Name	Length	Req'd	Data Type	Comments
ADJ_STG_UP_ID	12	Y	N	This must correspond with the prime key of the related CI_ADJ_STG_UP record.
CHAR_TYPE_CD	8	Y	A/N	This must correspond with a characteristic type that is defined as valid for `adjustment` Refer to Setting Up Characteristic Types & Their Values for more information.
SEQ_NUM	3	Y	N	This should be set to `10` unless you have multiple values for a given adjustment upload staging and characteristic type.
CHAR_VAL	16	N	A/N	Populate this field if your characteristic type is `predefined`.
ADHOC_CHAR_VAL	254	N	A/N	Populate this field if your characteristic type is `ad-hoc` or `file location`.
CHAR_VAL_FK1 - CHAR_VAL_FK5	50 each	N	A/N	Populate these fields if your characteristic type is `foreign key reference`. Up to five columns of 50 bytes each are provided to accommodate compound keys.

The Lifecycle of an Adjustment Staging Control Record

The following diagram shows the possible lifecycle of an adjustment staging control record.



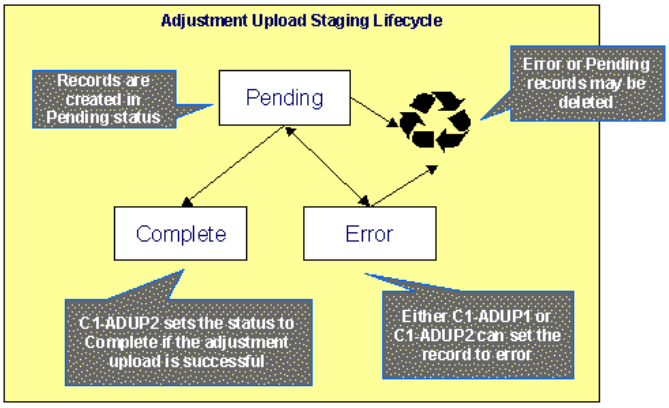
- `Pending`. An adjustment staging control record is created in this state. The `C1-ADUP1` process selects `pending` adjustment staging control records for validation.
- `In Progress`. The `C1-ADUP1` process sets a `pending` record to `in progress` when the totals on the adjustment staging control are successfully validated against the totals from the associated adjustment upload staging records.

Copyright © 2024, Oracle and/or its affiliates.

- `Complete`. The `C1-ADUP2` process sets the adjustment staging control's status to `complete` when all adjustment upload staging records linked to the adjustment staging control are `complete`.
- `Error`. The `C1-ADUP1` process sets a `pending` record to `error` if the adjustment staging control fails validation. The status may be set back to `pending` after the error is fixed.
- `Held`. The `held` status is available for situations where you want to prevent or delay the upload of a batch of adjustment staging records. The status may be set back to `pending` when the batch of records is ready for upload.
- `Pending`, `error` and `held` records may be deleted.

The Lifecycle of an Adjustment Upload Staging Record

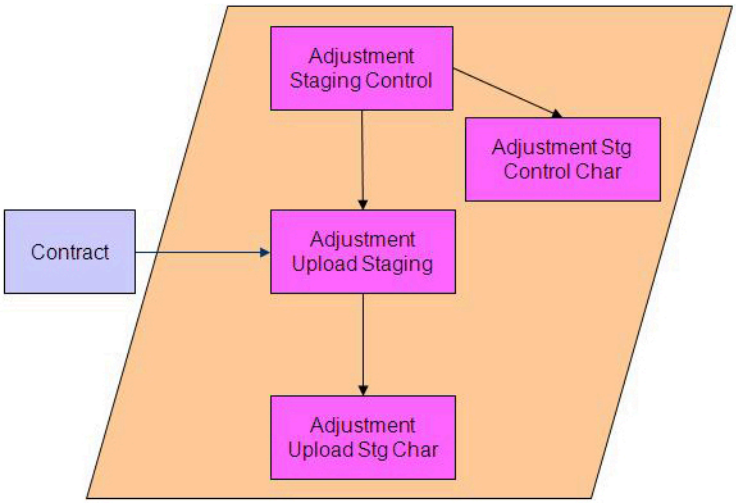
The following diagram shows the possible lifecycle of an adjustment upload staging record.



- `Pending`. Adjustment upload records are created in this state. The `C1-ADUP2` process selects `pending` adjustment upload records and creates adjustments for each of them.
- `Complete`. The `C1-ADUP2` process sets a `pending` record to `complete` if an adjustment is successfully created.
- `Error`. Either `C1-ADUP1` or `C1-ADUP2` process sets `pending` record to `error` when it encounters an error during the upload process. The status may be set back to `pending` after the error is fixed.
- `Pending` and `error` records may be deleted.

Process C1-ADUP1 - Preprocess Adjustment Uploads

The batch process identified by batch process ID `C1-ADUP1` refers to the background process that validates adjustment staging control records and populates contract IDs on adjustment upload staging records that do not specify an contract ID.



Copyright © 2024, Oracle and/or its affiliates.

Phase 1 - Validate Adjustment Staging Controls

The following points describe, at a high level, the first phase of the adjustment upload pre-process:

- For each `Pending` adjustment staging control,
-
  - Check that the record count on the adjustment staging control record equals the number of adjustment upload staging records that are linked to the adjustment staging control.
  - Check that the total adjustment amount on the adjustment staging control record equals the sum of the adjustment amounts from the adjustment upload staging records that are linked to the adjustment staging control.
  - If the adjustment staging control passes validation, set its status to `In Progress`. Otherwise, set the status to `Error`. Create a To Do entry using the inputs To Do type and To Do role (if supplied) for adjustment staging control errors. (Complete any outstanding To Do entries for the adjustment staging control before creating a new To Do entry.)
  - If no errors occur, perform To Do cleanup by completing any outstanding To Do entries that were previously created for the adjustment staging control.

Note: You can fix errors by going to the To Do entry and drilling into the adjustment staging control page. Don't forget to change the adjustment staging control's status back to `Pending` after fixing the error.

Phase 2 - Populate Contract ID

The following points describe, at a high level, the second phase of the adjustment upload pre-process:

- For each `Pending` adjustment upload staging that is linked to an `In Progress` adjustment staging control AND does not have the contract ID,
-
  - Execute the `Determine Contract` algorithm that is plugged in on the adjustment type.
  - If the algorithm returns a valid contract ID, stamp that contract ID onto the adjustment upload staging. If the algorithm also returns an indication that the adjustment is to be put into suspense, set the suspense flag on the adjustment upload staging to `In Suspense`. Refer to Suspense Adjustments for more information on how suspense adjustments are handled.
  - If the algorithm returns an error, set the adjustment upload staging record's status to `Error`. Create a To Do entry using the inputs To Do type and To Do role (if supplied) for adjustment upload staging errors. (Complete any outstanding To Do entries for the adjustment upload staging before creating a new To Do entry.)
  - If no errors occur, perform To Do cleanup by completing any outstanding To Do entries that were previously created for the adjustment upload staging.

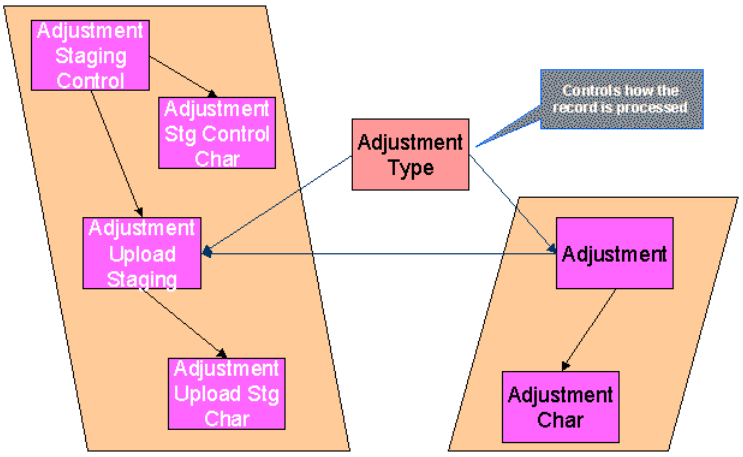
Note: You can fix errors by going to the To Do entry and drilling into the adjustment upload staging page. Don't forget to change the adjustment upload staging's status back to `Pending` after fixing the error.

Process C1-ADUP2 - Upload Adjustments

The batch process identified by batch process ID `C1-ADUP2` refers to the background process that creates adjustments for all adjustment upload staging records that are stamped with an contract ID.

The following diagram and section describe, at a high level, the processing done in the `C1-ADUP2` background process.

Copyright © 2024, Oracle and/or its affiliates.



- For each `Pending` adjustment upload staging that has an contract ID,
- • Add a frozen adjustment using the amount, creation date and adjustment type on the staging record.

> **Note:** If the type of adjustment being uploaded is one that is calculated, the algorithm that's plugged in on the adjustment type will calculate the adjustment amount and generate associated calculation lines (if applicable). See Calculated Adjustments for more information.

- If adjustment creation is successful, update the adjustment ID on the staging record and set the staging record's status to `Complete`.
- If adjustment creation results in an error, set the staging record's status to `Error`. Create a To Do entry using the To Do type on which this batch process (`C1-ADUP2`) is defined as creation process. (Complete any outstanding To Do entries for the adjustment upload staging before creating a new To Do entry.)
- If no errors occur, perform To Do cleanup by completing any outstanding To Do entries that were previously created for the adjustment upload staging.

Suspense Adjustments

What Are Suspense Adjustments?

When the adjustment upload preprocessor is unable to identify a valid contract ID, it can post the adjustment to a suspense contract. This is similar to how a payment is posted to a suspense account when a valid account ID could not be identified during payment upload.

Putting adjustments in suspense is optional. Therefore, this logic sits in a plug-in spot. If suspense adjustments are applicable to you, you must plug in your suspense logic in a `Determine Contract` algorithm on Adjustment Type.

How Are Suspense Adjustments Resolved?

A background process can be run periodically to automatically resolve suspense adjustments. Since rules for resolving suspense may vary, this logic sits in a plug-in spot. You must plug in your resolve logic in a `Resolve Suspense` algorithm on Adjustment Type.

- Process C1-ADURS - Resolve Suspense Adjustments on page 478

Process C1-ADURS - Resolve Suspense Adjustments

The batch process identified by batch process ID `C1-ADURS` refers to the background process that resolves suspense adjustments.

For each adjustment upload staging that is `In Suspense`, execute the `Resolve Suspense` algorithm that is plugged in on the adjustment type.

Copyright © 2024, Oracle and/or its affiliates.

Maintaining Adjustment Staging Control

Use this page to add, view or modify an adjustment staging control record. Open this page using **Billing Management, Adjustment Staging Control.**

Description of Page

Adjustment Staging Control ID is the unique identifier of the adjustment staging control record.

Create Date/Time is when the adjustment staging control was created. This field is protected after the record is saved.

Number of Staging Records is the number of adjustment upload staging records that are linked to this adjustment staging control record.

Total Adjustment Amount is the total of adjustment amounts from the adjustment upload staging records that are linked to this adjustment staging control record.

Enter the **Currency** for the total adjustment amount on the adjustment staging control. This is also the currency of the adjustment amounts on the associated adjustment upload staging records.

Status shows the state of the adjustment staging control. Possible values are: `Pending`, `In Progress`, `Complete`, `Error`, and `Held`.

A **Hold** button appears adjacent to the status description if **Status** is `Pending`. Click this button to set the status to `Held`. The adjustment upload process will ignore staging controls in this state.

A **Pend** button appears adjacent to the status description if **Status** is either `Held` or `Error`. Click this button to set the status to `Pending`. An error message box also appears below the status descriptionif **Status** is `Error`.

Use the **Characteristics** collection to capture miscellaneous information about an adjustment staging control.

- **Characteristic Type** The type of characteristic.
- **Characteristics Value** The value of the characteristic.

Note: You can only choose characteristic types defined as permissible on an adjustment staging control. Refer to Setting Up Characteristic Types & Their Values for more information.

Maintaining Adjustment Upload Staging

Use this page to add, view or modify an adjustment upload staging record. Open this page using **Billing Management, Adjustment Upload Staging.**

Description of Page

Adjustment Upload ID is the unique identifier of the adjustment upload staging record.

Adjustment Staging Control ID is the identifier of the adjustment staging control to which the adjustment upload staging is linked.

Indicate the **Adjustment Type**. This field is very important as it controls numerous aspects of the adjustment's impact on the customer's balance and your general ledger.

Note: You can only choose certain adjustment types. The contract's contract type has a collection of valid adjustment profiles. You may only reference adjustment types that are listed in one of the adjustment type profiles linked to the contract type.

Enter an **Adjustment Amount**.

Creation Date is the date when the adjustment occurred. This will be the creation date on the uploaded adjustment. This field is protected after the record is saved.

Contract indicates the contract to which the adjustment was posted. If the adjustment is in suspense, the label shows **Suspense Contract** instead.

Copyright © 2024, Oracle and/or its affiliates.

Adjustment is a reference to the adjustment that got created from the upload.

Status shows the state of the adjustment upload staging. Possible values are: `Pending`, `Complete`, and `Error`.

If **Status** is `Error`, a **Pend** button appears adjacent to the status description. In addition, an error message box displays below the status description.

If the adjustment is in suspense, a message appears to indicate the condition and the **Suspense Adjustment** is shown. A different message appears when the suspense is resolved.

Use the **Characteristics** collection to keep miscellaneous information about an adjustment upload staging.

- **Characteristic Type** The type of characteristic.
- **Characteristics Value** The value of the characteristic.

Note: An adjustment upload staging's characteristics default from the adjustment type.

Adjustment Creation, Cancellation, and View Through Inbound Web Service

Oracle Revenue Management and Billing enables you to create, cancel, and view adjustments through a web service. You can send the adjustment creation, adjustment cancellation, or adjustment view request from the external system through an inbound web service. The following three business services are shipped with the product to support different adjustment operations:

Business Service	Description
C1-CreateAdjustment	Used to create an adjustment against a contract. For more information, see C1-CreateAdjustment Business Service on page 482.
C1-CancelAdjustment	Used to cancel an adjustment. For more information, see C1-CancelAdjustment Business Service on page 498.
C1-GetAdjustmentHistory	Used to view the adjustments of an account. For more information, see C1-GetAdjustmentHistory Business Service on page 506.

You can create one inbound web service which supports all three operations - adjustment creation, adjustment cancellation, and adjustment view. Alternatively, you can create three inbound web services - one for adjustment creation, another for adjustment cancellation, and third for adjustment view.

You can create a SOAP or REST based inbound web service. If you want to send the request in the XML and/or JSON format, you need to ensure that you create a REST based inbound web service. However, if you want to send the request in the XML format, you can create a SOAP or REST based inbound web service.

For more information on how to setup this feature, see Prerequisites on page 480.

Prerequisites

The following table lists the prerequisites for adjustment creation, cancellation, and view through an inbound web service:

If you want to...	Then...
Create adjustments through an inbound web service	You need to do the following: • Define a REST or SOAP based inbound web service to call the **C1-CreateAdjustment** business service.

Copyright © 2024, Oracle and/or its affiliates.

If you want to...	Then...
	• Authorize the ORMB user to access the required application service. • Define the required account identifier types, divisions, contract types, and currencies in the system. • Define the required calculated or non-calculated adjustment types in the system. • Define the required characteristic types where the characteristic entity is set to **Adjustment**. • Associate the required characteristic types with the required adjustment types. • Include the adjustment type in the adjustment type profile which is associated with the respective contract type. • Define the required exchange rates in the system. • Define a currency conversion algorithm for adjustments using the **C1_CURALGAD** or **C1_EXCRTAD** algorithm type. • Attach a currency conversion algorithm to the **Currency Conversion For Adjustments** algorithm spot of the required divisions.
Cancel adjustments through an inbound web service	You need to do the following: • Define a REST or SOAP based inbound web service to call the **C1-CancelAdjustment** business service. • Authorize the ORMB user to access the required application service. • Define the required adjustment cancel reasons in the system.
View adjustments through an inbound web service	You need to do the following: • Define a REST or SOAP based inbound web service to call the **C1-GetAdjustmentHistory** business service. • Authorize the ORMB user to access the required application service. • Define the required account identifier types, divisions, contract types, and adjustment types in the system. • Set the **Set Limit for IWS Retrieving Records** option type of the **C1-RECLMT** feature configuration.
Enable the **Inbound Web Service History** feature	Refer to **Inbound Web Service History** section in the *Oracle Revenue Management and Billing Administrative Guide* to view the prerequisites.

Related Topics

For more information on...	See...
How to setup the **C1-RECLMT** feature configuration	Setting the C1-RECLMT Feature Configuration on page 1096

Copyright © 2024, Oracle and/or its affiliates.

C1-CreateAdjustment Business Service

The **Create Adjustment (C1-CreateAdjustment)** business service enables you to create an adjustment against a contract. It enables you to create both the calculated and non-calculated adjustments against a contract. You can call this business service from an external system through an inbound web service. If you want to send the adjustment creation request in the XML and/or JSON format, you need to ensure that you create a REST based inbound web service in Oracle Revenue Management and Billing. However, if you want to send the adjustment creation request in the XML format, you can create a SOAP or REST based inbound web service in Oracle Revenue Management and Billing.

You can specify either of the following match type while sending the adjustment creation request:

- **CNTY** - Used when you want to derive an active contract which is created using the specified contract type on the account. The system then adjusts the contract value of the account through the adjustment.

  Note: If there are multiple active contracts of the specified contract type on the account, the system will derive the contract with the latest start date.

- **CONT** - Used when you want to specify the contract whose value should be adjusted through the adjustment.

If the match type is set to **CNTY**, you need to specify either of the following in the adjustment creation request:

- Account ID
- Account Identifier Type, Account Identifier, and Division

However, if the match type is set to **CONT**, the system ignores the account information given in the adjustment creation request. On processing the adjustment creation request, the system creates an adjustment using the specified adjustment type. If an approval profile is set in the adjustment type, the system creates the adjustment in the **Freezable** status. It also creates an approval request for the adjustment. However, if an approval profile is not set in the adjustment type, the system creates the adjustment in the **Freezable** or **Frozen** status depending on the adjustment freeze option set in the adjustment type. Note that the system creates an adjustment against the contract when it is in the **Active** status.

You can specify the currency while sending the adjustment creation request. If the currency is not specified in the request, the system creates the adjustment in the account's invoice currency. However, if the specified currency is different from the account's invoice currency, the system does the currency conversion and creates the adjustment in the account's invoice currency.

The system enables you to track the adjustment creation response sent to the external system using the **Inbound Web Service History** feature.

For more information, see the **Inbound Web Service History** section in the *Oracle Revenue Management and Billing Administrative Guide*.

Related Topics

For more information on...	See...
Adjustment Creation Request in the XML Format	Adjustment Creation Request in the XML Format on page 483
Adjustment Creation Response in the XML Format	Adjustment Creation Response in the XML Format on page 483
Adjustment Creation Request in the JSON Format	Adjustment Creation Request in the JSON Format on page 484
Adjustment Creation Response in the JSON Format	Adjustment Creation Response in the JSON Format on page 485
Adjustment Creation Request and Response - Tags in XML Format	Adjustment Creation Request and Response - Tags in XML Format on page 486

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
Adjustment Creation Request and Response - Attribute/Value Pairs in JSON Format	Adjustment Creation Request and Response - Attribute/Value Pairs in JSON Format on page 491
Sample Adjustment Creation Request and Response in the XML Format	Sample Adjustment Creation Request and Response in the XML Format on page 496
Sample Adjustment Creation Request and Response in the JSON Format	Sample Adjustment Creation Request and Response in the JSON Format on page 497

Adjustment Creation Request in the XML Format

The following code snippet illustrates how the tags should be nested while receiving the adjustment creation request in the XML format:

```
<C1-CreateAdjustment>
    <adjustmentRequest>
        <accountNumber></accountNumber>
        <accountNumberTypeCd></accountNumberTypeCd>
        <division></division>
        <accountId></accountId>
        <matchTypeCd></matchTypeCd>
        <matchTypeValue></matchTypeValue>
        <comment></comment>
        <billId></billId>
        <currencyCd></currencyCd>
        <adjustmentType></adjustmentType>
        <currentAmount></currentAmount>
        <arrearsDate></arrearsDate>
        <accountingDate></accountingDate>
        <sourceSystem></sourceSystem>
        <externalTransactionId></externalTransactionId>
        <externalSourceId></externalSourceId>
        <adjustmentCharacteristics>
            <characteristicType></characteristicType>
            <characteristicValue></characteristicValue>
        </adjustmentCharacteristics>
    </adjustmentRequest>
</C1-CreateAdjustment>
```

Related Topics

For more information on...	See...
Adjustment Creation Response in the XML Format	Adjustment Creation Response in the XML Format on page 483
Adjustment Creation Request and Response - Tags in XML Format	Adjustment Creation Request and Response - Tags in XML Format on page 486
Sample Adjustment Creation Request and Response in the XML Format	Sample Adjustment Creation Request and Response in the XML Format on page 496

Adjustment Creation Response in the XML Format

The following code snippet illustrates how the tags are nested while sending the adjustment creation response in the XML format:

```
<C1-CreateAdjustment>
    <adjustmentRequest>
```

Copyright © 2024, Oracle and/or its affiliates.

```
        <accountNumber></accountNumber>
        <accountNumberTypeCd></accountNumberTypeCd>
        <division></division>
        <accountId></accountId>
        <matchTypeCd></matchTypeCd>
        <matchTypeValue></matchTypeValue>
        <comment></comment>
        <billId></billId>
        <currencyCd></currencyCd>
        <adjustmentType></adjustmentType>
        <currentAmount></currentAmount>
        <arrearsDate></arrearsDate>
        <accountingDate></accountingDate>
        <sourceSystem></sourceSystem>
        <externalTransactionId></externalTransactionId>
        <externalSourceId></externalSourceId>
        <adjustmentCharacteristics>
            <characteristicType></characteristicType>
            <characteristicValue></characteristicValue>
        </adjustmentCharacteristics>
    </adjustmentRequest>
    <adjustmentResponse>
        <adjustmentId></adjustmentId>
        <adjustmentStatus></adjustmentStatus>
        <saId></saId>
        <adjustmentAmount></adjustmentAmount>
        <approvalRequestId></approvalRequestId>
        <messageText></messageText>
        <statusCd></statusCd>
    </adjustmentResponse>
</C1-CreateAdjustment>
```

Related Topics

For more information on...	See...
Adjustment Creation Request in the XML Format	Adjustment Creation Request in the XML Format on page 483
Adjustment Creation Request and Response - Tags in XML Format	Adjustment Creation Request and Response - Tags in XML Format on page 486
Sample Adjustment Creation Request and Response in the XML Format	Sample Adjustment Creation Request and Response in the XML Format on page 496

Adjustment Creation Request in the JSON Format

The following code snippet illustrates how the attribute/value pairs should be structured while receiving the adjustment creation request in the JSON format:

```
{
    -"C1-CreateAdjustment": {
        -"adjustmentRequest": {
            -"accountNumber": -"",
            -"accountNumberTypeCd": -"",
            -"division": -"",
            -"accountId": -"",
            -"matchTypeCd": -"",
            -"matchTypeValue": -"",
            -"comment": -"",
            -"billId": -"",
            -"currencyCd": -"",
            -"adjustmentType": -"",
```

Copyright © 2024, Oracle and/or its affiliates.

```
            -"currentAmount": -"",
            -"arrearsDate": -"",
            -"accountingDate": -"",
            -"sourceSystem": -"",
            -"externalTransactionId": -"",
            -"externalSourceId": -"",
            -"adjustmentCharacteristics": {
                -"characteristicType": -"",
                -"characteristicValue": -""
            -}
        -}
    -}
}
```

Related Topics

For more information on...	See...
Adjustment Creation Response in the JSON Format	Adjustment Creation Response in the JSON Format on page 485
Adjustment Creation Request and Response - Attribute/Value Pairs in JSON Format	Adjustment Creation Request and Response - Attribute/Value Pairs in JSON Format on page 491
Sample Adjustment Creation Request and Response in the JSON Format	Sample Adjustment Creation Request and Response in the JSON Format on page 497

Adjustment Creation Response in the JSON Format

The following code snippet illustrates how the attribute/value pairs are structured while sending the adjustment creation response in the JSON format:

```
{
    -"C1-CreateAdjustment": {
        -"adjustmentRequest": {
            -"accountNumber": -"",
            -"accountNumberTypeCd": -"",
            -"division": -"",
            -"accountId": -"",
            -"matchTypeCd": -"",
            -"matchTypeValue": -"",
            -"comment": -"",
            -"billId": -"",
            -"currencyCd": -"",
            -"adjustmentType": -"",
            -"currentAmount": -"",
            -"arrearsDate": -"",
            -"accountingDate": -"",
            -"sourceSystem": -"",
            -"externalTransactionId": -"",
            -"externalSourceId": -"",
            -"adjustmentCharacteristics": {
                -"characteristicType": -"",
                -"characteristicValue": -""
            -}
        -}
        -"adjustmentResponse": {
            -"adjustmentId": -"",
            -"adjustmentStatus": -"",
            -"saId": -"",
            -"adjustmentAmount": -"",
            -"approvalRequestId": -"",
```

Copyright © 2024, Oracle and/or its affiliates.

```
            -"messageText": -"",
            -"statusCd": -""
        -}
    -}
}
```

Related Topics

For more information on...	See...
Adjustment Creation Request in the JSON Format	Adjustment Creation Request in the JSON Format on page 484
Adjustment Creation Request and Response - Attribute/Value Pairs in JSON Format	Adjustment Creation Request and Response - Attribute/Value Pairs in JSON Format on page 491
Sample Adjustment Creation Request and Response in the JSON Format	Sample Adjustment Creation Request and Response in the JSON Format on page 497

Adjustment Creation Request and Response - Tags in XML Format

Note: We recommend you to refer the Adjustment Creation Request in the XML Format on page 483 and Adjustment Creation Response in the XML Format on page 483 topics in parallel while understanding the below mentioned tags. This will help you to understand how the tags are nested in the XML format.

Before calling the **C1-CreateAdjustment** business service through an inbound web service, you need to ensure that the adjustment creation request contains the following tags:

Tag Name	Tag Description	Mandatory (Yes or No)
C1-CreateAdjustment	Used to indicate that you want to invoke the **C1-CreateAdjustment** business service.	Yes
adjustmentRequest	Used to specify the details of the adjustment.	Yes
accountNumber	Used to indicate the account for which you want to create the adjustment.	Yes (Conditional) **Note:** This data is required when: • **CNTY** is specified in the **matchTypeCd** tag. • The value is not specified in the **accountId** tag.
accountNumberTypeCd	Used to specify the account identifier type. **Note:** You must specify an account identifier type which is already defined in the system.	Yes (Conditional) **Note:** This data is required when: • **CNTY** is specified in the **matchTypeCd** tag. • The value is not specified in the **accountId** tag.
division	Used to indicate the division to which the account belongs.	Yes (Conditional)

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Tag Description	Mandatory (Yes or No)
	Note: You must specify a division which is already defined in the system.	**Note:** This data is required when: • **CNTY** is specified in the **matchTypeCd** tag. • The value is not specified in the **accountId** tag.
accountId	Used to indicate the account for which you want to create the adjustment.	Yes (Conditional) **Note:** This data is required when: • **CNTY** is specified in the **matchTypeCd** tag. • The values are not specified in the **accountNumber**, **accountNumberTypeCd**, and **division** tags.
matchTypeCd	Used to indicate the match type using which you want to identify the contract against which the adjustment should be created. The valid values are: • **CNTY** - Used when you want to derive an active contract which is created using the specified contract type on the account. The system then adjusts the contract value of the account through the adjustment. **Note:** If there are multiple active contracts of the specified contract type on the account, the system will derive the contract with the latest start date. • **CONT** - Used when you want to specify the contract whose value should be adjusted through the adjustment.	Yes
matchTypeValue	If **CNTY** is specified in the **matchTypeCd** tag, this tag is used to specify the contract type. However, if **CONT** is specified in the **matchTypeCd** tag, this tag is used to specify the contract ID. **Note:** You must specify a contract type or contract which is already defined in the system. You cannot specify multiple values separated by delimiter for a match type.	Yes
comment	Used to specify additional information about the adjustment.	No
billId	Used to indicate the previous completed bill against which you want to create the adjustment.	No

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Tag Description	Mandatory (Yes or No)
	Note: You must specify a bill which already exists in the system. The bill and contract should belong to the same account. The system allows you to create an adjustment against a bill when the **Print by Default** option is selected in the adjustment type.	
currencyCd	Used to indicate the currency in which you want to create the adjustment. **Note:** You must specify a currency which is already defined in the system. If you do not specify the currency, by default, the adjustment is created in the account's invoice currency. If the specified currency is different from the account's invoice currency, the system does the currency conversion and creates the adjustment in the account's invoice currency.	No
adjustmentType	Used to indicate the adjustment type using which you want to create the adjustment. **Note:** You must specify an adjustment type which is already defined in the system. You must ensure that the adjustment type is included in the adjustment type profile which is associated with the contract type.	Yes
currentAmount	Used to specify the adjustment amount. **Note:** If you do not specify the amount, by default, it is set to zero(0). The adjustment amount's precision must match the precision (decimal positions) defined for the currency.	No
arrearsDate	Used to specify the date from when the adjustment's financial transaction should start aging.	No

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Tag Description	Mandatory (Yes or No)
	Note: You must specify the date in the YYYY-MM-DD format. If you do not specify the date, by default, it is set to the date when the adjustment's financial transaction is created.	
accountingDate	Used to specify the date using which you want the system to determine the financial transaction's accounting period in the general ledger. **Note:** You must specify the date in the YYYY-MM-DD format. If you do not specify the date, by default, it is set to the date when the adjustment's financial transaction is created.	No
sourceSystem	Used to indicate the external system from which the adjustment creation request is sent. **Note:** You must specify an external system which is already defined in the system. The external system is used when the **Inbound Web Service History** feature is enabled. The specified external system should match the external system which is associated with the outbound message type specified in the **C1-IWSHIST** algorithm. If the external system is not specified in the adjustment creation request, the system, by default, stamps the external system associated with the outbound message type (specified in the **C1-IWSHIST** algorithm) corresponding to the record in the **C1_IWS_HIST** table. For more information, see the **Inbound Web Service History** section in the *Oracle Revenue Management and Billing Administrative Guide*.	No
externalTransactionId	Used to indicate the transaction in the external system through which the adjustment creation request is sent. **Note:** The external transaction ID is used when the **Inbound Web Service History** feature is enabled.	No
externalSourceId	Used to specify the external source system ID.	No

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Tag Description	Mandatory (Yes or No)
	Note: The external source system ID is used when the **Inbound Web Service History** feature is enabled.	
adjustmentCharacteristics	Used to specify a characteristic for the adjustment.	No
characteristicType	Used to indicate the characteristic type.	Yes (Conditional)
	Note: You must specify a characteristic type where the characteristic entity is set to **Adjustment** and which is associated with the adjustment type.	**Note:** This data is required while defining a characteristic for the adjustment.
characteristicValue	Used to specify the value for the characteristic type.	Yes (Conditional)
		Note: This data is required while defining a characteristic for the adjustment.

The following table lists and describes the tags which appear in the adjustment creation response in the XML format:

Tag Name	Tag Description
adjustmentResponse	Indicates whether the adjustment creation request is successfully processed or not.
adjustmentId	Displays the adjustment ID.
adjustmentStatus	Indicates the status of the adjustment. The valid values are: • Freezable • Frozen The following table indicates the scenarios in which the adjustment is set to the **Freezable** or **Frozen** status:

If...	Then...
An approval profile is specified in the adjustment type	The adjustment status is set to **Freezable**.
An approval profile is not specified in the adjustment type, but the adjustment freeze option in the adjustment type is set to **Freeze At Bill Completion**	The adjustment status is set to **Freezable**.
An approval profile is not specified in the adjustment type, but the adjustment freeze option in the adjustment type is set to **Freeze At Will**	The adjustment status is set to **Frozen**.

saId	Indicates the contract against which the adjustment is created.
adjustmentAmount	Displays the adjustment amount.
approvalRequestId	Displays the approval request ID.

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Tag Description
	Note: The data appears in this tag when the approval profile is set in the adjustment type using which the adjustment is created.
messageText	Indicates the error that occurred while creating the adjustment.
	Note: This tag appears when the value in the **statusCd** tag is set to **Fail**.
statusCd	Indicates whether the adjustment creation request is successfully processed or not. The valid values are: • Success • Fail

Related Topics

For more information on...	See...
Sample Adjustment Creation Request and Response in the XML Format	Sample Adjustment Creation Request and Response in the XML Format on page 496

Adjustment Creation Request and Response - Attribute/Value Pairs in JSON Format

Note: We recommend you to refer the Adjustment Creation Request in the JSON Format on page 484 and Adjustment Creation Response in the JSON Format on page 485 topics in parallel while understanding the below mentioned attributes. This will help you to understand how the attribute/value pairs are structured in the JSON format.

Before calling the **C1-CreateAdjustment** business service through an inbound web service, you need to ensure that the adjustment creation request contains the following attributes:

Attribute Name	Attribute Description	Mandatory (Yes or No)
C1-CreateAdjustment	Used to indicate that you want to invoke the **C1-CreateAdjustment** business service.	Yes
adjustmentRequest	Used to specify the details of the adjustment.	Yes
accountNumber	Used to indicate the account for which you want to create the adjustment.	Yes (Conditional)
		Note: This data is required when: • The **matchTypeCd** attribute is set to **CNTY**. • No value is specified for the **accountId** attribute.
accountNumberTypeCd	Used to specify the account identifier type.	Yes (Conditional)
	Note: You must specify an account identifier type which is already defined in the system.	**Note:** This data is required when: • The **matchTypeCd** attribute is set to **CNTY**. • No value is specified for the **accountId** attribute.

Copyright © 2024, Oracle and/or its affiliates.

Attribute Name	Attribute Description	Mandatory (Yes or No)
division	Used to indicate the division to which the account belongs. **Note:** You must specify a division which is already defined in the system.	Yes (Conditional) **Note:** This data is required when: • The **matchTypeCd** attribute is set to **CNTY**. • No value is specified for the **accountId** attribute.
accountId	Used to indicate the account for which you want to create the adjustment.	Yes (Conditional) **Note:** This data is required when: • The **matchTypeCd** attribute is set to **CNTY**. • No value is specified for the **accountNumber**, **accountNumberTypeCd**, and **division** attributes.
matchTypeCd	Used to indicate the match type using which you want to identify the contract against which the adjustment should be created. The valid values are: • **CNTY** - Used when you want to derive an active contract which is created using the specified contract type on the account. The system then adjusts the contract value of the account through the adjustment. **Note:** If there are multiple active contracts of the specified contract type on the account, the system will derive the contract with the latest start date. • **CONT** - Used when you want to specify the contract whose value should be adjusted through the adjustment.	Yes
matchTypeValue	If the **matchTypeCd** attribute is set to **CNTY**, this attribute is used to specify the contract type. However, if the **matchTypeCd** attribute is set to **CONT**, this attribute is used to specify the contract ID. **Note:** You must specify a contract type or contract which is already defined in the system.	Yes
comment	Used to specify additional information about the adjustment.	No
billId	Used to indicate the previous completed bill against which you want to create the adjustment.	No

Copyright © 2024, Oracle and/or its affiliates.

Attribute Name	Attribute Description	Mandatory (Yes or No)
	Note: You must specify a bill which already exists in the system. The bill and contract should belong to the same account. The system allows you to create an adjustment against a bill when the **Print by Default** option is selected in the adjustment type.	
currencyCd	Used to indicate the currency in which you want to create the adjustment. **Note:** You must specify a currency which is already defined in the system. If you do not specify the currency, by default, the adjustment is created in the account's invoice currency. If the specified currency is different from the account's invoice currency, the system does the currency conversion and creates the adjustment in the account's invoice currency.	No
adjustmentType	Used to indicate the adjustment type using which you want to create the adjustment. **Note:** You must specify an adjustment type which is already defined in the system. You must ensure that the adjustment type is included in the adjustment type profile which is associated with the contract type.	Yes
currentAmount	Used to specify the adjustment amount. **Note:** If you do not specify the amount, by default, it is set to zero(0). The adjustment amount's precision must match the precision (decimal positions) defined for the currency.	No
arrearsDate	Used to specify the date from when the adjustment's financial transaction should start aging.	No

Copyright © 2024, Oracle and/or its affiliates.

Attribute Name	Attribute Description	Mandatory (Yes or No)
	Note: If you do not specify the date, by default, it is set to the date when the adjustment's financial transaction is created.	
accountingDate	Used to specify the date using which you want the system to determine the financial transaction's accounting period in the general ledger. **Note:** If you do not specify the date, by default, it is set to the date when the adjustment's financial transaction is created.	No
sourceSystem	Used to indicate the external system from which the adjustment creation request is sent. **Note:** You must specify an external system which is already defined in the system. The external system is used when the **Inbound Web Service History** feature is enabled. The specified external system should match the external system which is associated with the outbound message type specified in the **C1-IWSHIST** algorithm. If the external system is not specified in the adjustment creation request, the system, by default, stamps the external system associated with the outbound message type (specified in the **C1-IWSHIST** algorithm) corresponding to the record in the **C1_IWS_HIST** table. For more information, see the **Inbound Web Service History** section in the *Oracle Revenue Management and Billing Administrative Guide*.	No
externalTransactionId	Used to indicate the transaction in the external system through which the adjustment creation request is sent. **Note:** The external transaction ID is used when the **Inbound Web Service History** feature is enabled.	No
externalSourceId	Used to specify the external source system ID. **Note:** The external source system ID is used when the **Inbound Web Service History** feature is enabled.	No
adjustmentCharacteristics	Used to specify a characteristic for the adjustment.	No
characteristicType	Used to indicate the characteristic type.	Yes (Conditional)

Copyright © 2024, Oracle and/or its affiliates.

Attribute Name	Attribute Description	Mandatory (Yes or No)
	Note: You must specify a characteristic type where the characteristic entity is set to **Adjustment** and which is associated with the adjustment type.	**Note:** This data is required while defining a characteristic for the adjustment.
characteristicValue	Used to specify the value for the characteristic type.	Yes (Conditional) **Note:** This data is required while defining a characteristic for the adjustment.

The following table lists and describes the attributes which appear in the adjustment creation response in the JSON format:

Attribute Name	Attribute Description
adjustmentResponse	Indicates whether the adjustment creation request is successfully processed or not.
adjustmentId	Displays the adjustment ID.
adjustmentStatus	Indicates the status of the adjustment. The valid values are: • Freezable • Frozen The following table indicates the scenarios in which the adjustment is set to the **Freezable** or **Frozen** status: **If...** / **Then...** An approval profile is specified in the adjustment type / The adjustment status is set to **Freezable**. An approval profile is not specified in the adjustment type, but the adjustment freeze option in the adjustment type is set to **Freeze At Bill Completion** / The adjustment status is set to **Freezable**. An approval profile is not specified in the adjustment type, but the adjustment freeze option in the adjustment type is set to **Freeze At Will** / The adjustment status is set to **Frozen**.
saId	Indicates the contract against which the adjustment is created.
adjustmentAmount	Displays the adjustment amount.
approvalRequestId	Displays the approval request ID. **Note:** The value for this attribute appears when the approval profile is set in the adjustment type using which the adjustment is created.
messageText	Indicates the error that occurred while creating the adjustment. **Note:** This attribute appears when the value of the **statusCd** attribute is set to **Fail**.

Copyright © 2024, Oracle and/or its affiliates.

Attribute Name	Attribute Description
statusCd	Indicates whether the adjustment creation request is successfully processed or not. The valid values are: • Success • Fail

Related Topics

For more information on...	See...
Sample Adjustment Creation Request and Response in the JSON Format	Sample Adjustment Creation Request and Response in the JSON Format on page 497

Sample Adjustment Creation Request and Response in the XML Format

Using the below example, you can create an adjustment for the 7895678365 bill against the 3455678937 contract.

Sample Adjustment Creation Request in the XML Format

```
<C1-CreateAdjustment>
    <adjustmentRequest>
        <accountNumber></accountNumber>
        <accountNumberTypeCd></accountNumberTypeCd>
        <division></division>
        <accountId>5467878999</accountId>
        <matchTypeCd>CONT</matchTypeCd>
        <matchTypeValue>3455678937</matchTypeValue>
        <comment></comment>
        <billId>7895678365</billId>
        <currencyCd>USD</currencyCd>
        <adjustmentType>ADTY1</adjustmentType>
        <currentAmount>10.00</currentAmount>
        <arrearsDate>2020-10-02</arrearsDate>
        <accountingDate>2020-10-02</accountingDate>
        <sourceSystem>ES01</sourceSystem>
        <externalTransactionId>786790</externalTransactionId>
        <externalSourceId>2341</externalSourceId>
        <adjustmentCharacteristics>
            <characteristicType>CL01</characteristicType>
            <characteristicValue>20</characteristicValue>
        </adjustmentCharacteristics>
    </adjustmentRequest>
</C1-CreateAdjustment>
```

Sample Adjustment Creation Response in the XML Format

```
<C1-CreateAdjustment>
    <adjustmentRequest>
        <accountNumber></accountNumber>
        <accountNumberTypeCd></accountNumberTypeCd>
        <division></division>
        <accountId>5467878999</accountId>
        <matchTypeCd>CONT</matchTypeCd>
        <matchTypeValue>3455678937</matchTypeValue>
        <comment></comment>
        <billId>7895678365</billId>
        <currencyCd>USD</currencyCd>
        <adjustmentType>ADTY1</adjustmentType>
        <currentAmount>10.00</currentAmount>
```

Copyright © 2024, Oracle and/or its affiliates.

```
        <arrearsDate>2020-10-02</arrearsDate>
        <accountingDate>2020-10-02</accountingDate>
        <sourceSystem>ES01</sourceSystem>
        <externalTransactionId>786790</externalTransactionId>
        <externalSourceId>2341</externalSourceId>
        <adjustmentCharacteristics>
            <characteristicType>CL01</characteristicType>
            <characteristicValue>20</characteristicValue>
        </adjustmentCharacteristics>
    </adjustmentRequest>
    <adjustmentResponse>
        <adjustmentId>145678789723</adjustmentId>
        <adjustmentStatus>Frozen</adjustmentStatus>
        <saId>3455678937</saId>
        <adjustmentAmount>10.00</adjustmentAmount>
        <approvalRequestId></approvalRequestId>
        <statusCd>Success</statusCd>
    </adjustmentResponse>
</C1-CreateAdjustment>
```

Note: We have given the above sample to illustrate how the tags should be nested while sending an adjustment creation request from the external system. However, you should not seamlessly use this sample in your environment because the required pre-requisite data may not be available in your environment. In such case, the system will give erroneous results.

Related Topics

For more information on...	See...
Adjustment Creation Request and Response - Tags in XML Format	Adjustment Creation Request and Response - Tags in XML Format on page 486

Sample Adjustment Creation Request and Response in the JSON Format

Using the below example, you can create an adjustment against an active contract which is created using the CT01 contract type for the account.

Sample Adjustment Creation Request in the JSON Format

```
{
    -"C1-CreateAdjustment": {
        -"adjustmentRequest": {
            -"accountNumber": -"2345678900",
            -"accountNumberTypeCd": -"ACCTNUM",
            -"division": -"D0307",
            -"accountId": -"",
            -"matchTypeCd": -"CNTY",
            -"matchTypeValue": -"CT01",
            -"comment": -"",
            -"billId": -"",
            -"currencyCd": -"USD",
            -"adjustmentType": -"ADTY5",
            -"currentAmount": -"25.00",
            -"arrearsDate": -"",
            -"accountingDate": -"",
            -"sourceSystem": -"ES01",
            -"externalTransactionId": -"565678",
            -"externalSourceId": -"2341",
            -"adjustmentCharacteristics": {
                -"characteristicType": -"CL01",
                -"characteristicValue": -"50"
            -}
        -}
```

Copyright © 2024, Oracle and/or its affiliates.

```
    -}
}
```

Sample Adjustment Creation Response in the JSON Format

```
{
    -"C1-CreateAdjustment": {
        -"adjustmentRequest": {
            -"accountNumber": -"2345678900",
            -"accountNumberTypeCd": -"ACCTNUM",
            -"division": -"D0307",
            -"accountId": -"",
            -"matchTypeCd": -"CNTY",
            -"matchTypeValue": -"CT01",
            -"comment": -"",
            -"billId": -"",
            -"currencyCd": -"USD",
            -"adjustmentType": -"ADTY5",
            -"currentAmount": -"25.00",
            -"arrearsDate": -"",
            -"accountingDate": -"",
            -"sourceSystem": -"ES01",
            -"externalTransactionId": -"565678",
            -"externalSourceId": -"2341",
            -"adjustmentCharacteristics": {
                -"characteristicType": -"CL01",
                -"characteristicValue": -"50"
            -}
        -}
        -"adjustmentResponse": {
            -"adjustmentId": -"763456980012",
            -"adjustmentStatus": -"Freezable",
            -"saId": -"6756789012",
            -"adjustmentAmount": -"25.00",
            -"approvalRequestId": -"234567890123",
            -"statusCd": -"Success"
        -}
    -}
}
```

Note: We have given the above sample to illustrate how the attribute value pairs should be structured while sending an adjustment creation request from the external system. However, you should not seamlessly use this sample in your environment because the required pre-requisite data may not be available in your environment. In such case, the system will give erroneous results.

Related Topics

For more information on...	See...
Adjustment Creation Request and Response - Attribute/Value Pairs in JSON Format	Adjustment Creation Request and Response - Attribute/Value Pairs in JSON Format on page 491

C1-CancelAdjustment Business Service

The **Cancel Adjustment (C1-CancelAdjustment)** business service enables you to cancel an adjustment. You can call this business service from an external system through an inbound web service. If you want to send the adjustment cancellation request in the XML and/or JSON format, you need to ensure that you create a REST based inbound web service in Oracle Revenue Management and Billing. However, if you want to send the adjustment cancellation request

Copyright © 2024, Oracle and/or its affiliates.

in the XML format, you can create a SOAP or REST based inbound web service in Oracle Revenue Management and Billing.

You need to specify an adjustment cancel reason in the adjustment cancellation request. On processing the adjustment cancellation request, the system cancels the adjustment and changes the status of the adjustment to **Canceled**.

The system enables you to track the adjustment cancellation response sent to the external system using the **Inbound Web Service History** feature.

For more information, see the **Inbound Web Service History** section in the *Oracle Revenue Management and Billing Administrative Guide*.

Related Topics

For more information on...	See...
Adjustment Cancellation Request in the XML Format	Adjustment Cancellation Request in the XML Format on page 499
Adjustment Cancellation Response in the XML Format	Adjustment Cancellation Response in the XML Format on page 500
Adjustment Cancellation Request in the JSON Format	Adjustment Cancellation Request in the JSON Format on page 500
Adjustment Cancellation Response in the JSON Format	Adjustment Cancellation Response in the JSON Format on page 501
Adjustment Cancellation Request and Response - Tags in XML Format	Adjustment Cancellation Request and Response - Tags in XML Format on page 501
Adjustment Cancellation Request and Response - Attribute/Value Pairs in JSON Format	Adjustment Cancellation Request and Response - Attribute/Value Pairs in JSON Format on page 503
Sample Adjustment Cancellation Request and Response in the XML Format	Sample Adjustment Cancellation Request and Response in the XML Format on page 505
Sample Adjustment Cancellation Request and Response in the JSON Format	Sample Adjustment Cancellation Request and Response in the JSON Format on page 505

Adjustment Cancellation Request in the XML Format

The following code snippet illustrates how the tags should be nested while receiving the adjustment cancellation request in the XML format:

```
<C1-CancelAdjustment>
    <cancelAdjRequest>
        <adjustmentId></adjustmentId>
        <reason></reason>
        <sourceSystem></sourceSystem>
        <externalTransactionId></externalTransactionId>
        <externalSourceId></externalSourceId>
    </cancelAdjRequest>
</C1-CancelAdjustment>
```

Related Topics

For more information on...	See...
Adjustment Cancellation Response in the XML Format	Adjustment Cancellation Response in the XML Format on page 500

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
Adjustment Cancellation Request and Response - Tags in XML Format	Adjustment Cancellation Request and Response - Tags in XML Format on page 501
Sample Adjustment Cancellation Request and Response in the XML Format	Sample Adjustment Cancellation Request and Response in the XML Format on page 505

Adjustment Cancellation Response in the XML Format

The following code snippet illustrates how the tags are nested while sending the adjustment cancellation response in the XML format:

```
<C1-CancelAdjustment>
    <cancelAdjRequest>
        <adjustmentId></adjustmentId>
        <reason></reason>
        <sourceSystem></sourceSystem>
        <externalTransactionId></externalTransactionId>
        <externalSourceId></externalSourceId>
    </cancelAdjRequest>
    <cancelAdjResponse>
        <messageText></messageText>
        <cancelReasonDesc></cancelReasonDesc>
        <adjustmentStatus></adjustmentStatus>
        <statusCd></statusCd>
    </cancelAdjResponse>
</C1-CancelAdjustment>
```

Related Topics

For more information on...	See...
Adjustment Cancellation Request in the XML Format	Adjustment Cancellation Request in the XML Format on page 499
Adjustment Cancellation Request and Response - Tags in XML Format	Adjustment Cancellation Request and Response - Tags in XML Format on page 501
Sample Adjustment Cancellation Request and Response in the XML Format	Sample Adjustment Cancellation Request and Response in the XML Format on page 505

Adjustment Cancellation Request in the JSON Format

The following code snippet illustrates how the attribute/value pairs should be structured while receiving the adjustment cancellation request in the JSON format:

```
{
    -"C1-CancelAdjustment": {
        -"cancelAdjRequest": {
            -"adjustmentId": -"",
            -"reason": -"",
            -"sourceSystem": -"",
            -"externalTransactionId": -"",
            -"externalSourceId": -""
        -}
    -}
}
```

Related Topics

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
Adjustment Cancellation Response in the JSON Format	Adjustment Cancellation Response in the JSON Format on page 501
Adjustment Cancellation Request and Response - Attribute/Value Pairs in JSON Format	Adjustment Cancellation Request and Response - Attribute/Value Pairs in JSON Format on page 503
Sample Adjustment Cancellation Request and Response in the JSON Format	Sample Adjustment Cancellation Request and Response in the JSON Format on page 505

Adjustment Cancellation Response in the JSON Format

The following code snippet illustrates how the attribute/value pairs are structured while sending the adjustment cancellation response in the JSON format:

```
{
    -"C1-CancelAdjustment": {
        -"cancelAdjRequest": {
            -"adjustmentId": -"",
            -"reason": -"",
            -"sourceSystem": -"",
            -"externalTransactionId": -"",
            -"externalSourceId": -""
        -}
        -"cancelAdjResponse": {
            -"messageText": -"",
            -"cancelReasonDesc": -"",
            -"adjustmentStatus": -"",
            -"statusCd": -""
        -}
    -}
}
```

Related Topics

For more information on...	See...
Adjustment Cancellation Request in the JSON Format	Adjustment Cancellation Request in the JSON Format on page 500
Adjustment Cancellation Request and Response - Attribute/Value Pairs in JSON Format	Adjustment Cancellation Request and Response - Attribute/Value Pairs in JSON Format on page 503
Sample Adjustment Cancellation Request and Response in the JSON Format	Sample Adjustment Cancellation Request and Response in the JSON Format on page 505

Adjustment Cancellation Request and Response - Tags in XML Format

Note: We recommend you to refer the Adjustment Cancellation Request in the XML Format on page 499 and Adjustment Cancellation Response in the XML Format on page 500 topics in parallel while understanding the below mentioned tags. This will help you to understand how the tags are nested in the XML format.

Before calling the **C1-CancelAdjustment** business service through an inbound web service, you need to ensure that the adjustment cancellation request contains the following tags:

Tag Name	Tag Description	Mandatory (Yes or No)
C1-CancelAdjustment	Used to indicate that you want to invoke the **C1-CancelAdjustment** business service.	Yes

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Tag Description	Mandatory (Yes or No)
cancelAdjRequest	Used to specify the details of the adjustment.	Yes
adjustmentId	Used to indicate the adjustment that you want to cancel. **Note:** You must specify an adjustment which already exists in the system.	Yes
reason	Used to indicate the reason why you want to cancel the adjustment. **Note:** You must specify an adjustment cancel reason which is already defined in the system.	Yes
sourceSystem	Used to indicate the external system from which the adjustment cancellation request is sent. **Note:** You must specify an external system which is already defined in the system. The external system is used when the **Inbound Web Service History** feature is enabled. The specified external system should match the external system which is associated with the outbound message type specified in the **C1-IWSHIST** algorithm. If the external system is not specified in the adjustment cancellation request, the system, by default, stamps the external system associated with the outbound message type (specified in the **C1-IWSHIST** algorithm) corresponding to the record in the **C1_IWS_HIST** table. For more information, see the **Inbound Web Service History** section in the *Oracle Revenue Management and Billing Administrative Guide*.	No
externalTransactionId	Used to indicate the transaction in the external system through which the adjustment cancellation request is sent. **Note:** The external transaction ID is used when the **Inbound Web Service History** feature is enabled.	No
externalSourceId	Used to specify the external source system ID. **Note:** The external source system ID is used when the **Inbound Web Service History** feature is enabled.	No

The following table lists and describes the tags which appear in the adjustment cancellation response in the XML format:

Tag Name	Tag Description
cancelAdjResponse	Indicates whether the adjustment cancellation request is successfully processed or not.
messageText	Indicates the error that occurred while canceling the adjustment. **Note:** This tag appears when the value in the **statusCd** tag is set to **Fail**.

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Tag Description
cancelReasonDesc	Indicates the reason why the adjustment is canceled.
adjustmentStatus	Indicates the status of the adjustment. The valid value is: • Canceled
statusCd	Indicates whether the adjustment cancellation request is successfully processed or not. The valid values are: • Success • Fail

Related Topics

For more information on...	See...
Sample Adjustment Cancellation Request and Response in the XML Format	Sample Adjustment Cancellation Request and Response in the XML Format on page 505

Adjustment Cancellation Request and Response - Attribute/Value Pairs in JSON Format

Note: We recommend you to refer the Adjustment Cancellation Request in the JSON Format on page 500 and Adjustment Cancellation Response in the JSON Format on page 501 topics in parallel while understanding the below mentioned attributes. This will help you to understand how the attribute/value pairs are structured in the JSON format.

Before calling the **C1-CancelAdjustment** business service through an inbound web service, you need to ensure that the adjustment cancellation request contains the following attributes:

Attribute Name	Attribute Description	Mandatory (Yes or No)
C1-CancelAdjustment	Used to indicate that you want to invoke the **C1-CancelAdjustment** business service.	Yes
cancelAdjRequest	Used to specify the details of the adjustment.	Yes
adjustmentId	Used to indicate the adjustment that you want to cancel. **Note:** You must specify an adjustment which already exists in the system.	Yes
reason	Used to indicate the reason why you want to cancel the adjustment. **Note:** You must specify an adjustment cancel reason which is already defined in the system.	Yes
sourceSystem	Used to indicate the external system from which the adjustment cancellation request is sent.	No

Copyright © 2024, Oracle and/or its affiliates.

Attribute Name	Attribute Description	Mandatory (Yes or No)
	Note: You must specify an external system which is already defined in the system. The external system is used when the **Inbound Web Service History** feature is enabled. The specified external system should match the external system which is associated with the outbound message type specified in the **C1-IWSHIST** algorithm. If the external system is not specified in the adjustment cancellation request, the system, by default, stamps the external system associated with the outbound message type (specified in the **C1-IWSHIST** algorithm) corresponding to the record in the **C1_IWS_HIST** table. For more information, see the **Inbound Web Service History** section in the *Oracle Revenue Management and Billing Administrative Guide*.	
externalTransactionId	Used to indicate the transaction in the external system through which the adjustment cancellation request is sent. **Note:** The external transaction ID is used when the **Inbound Web Service History** feature is enabled.	No
externalSourceId	Used to specify the external source system ID. **Note:** The external source system ID is used when the **Inbound Web Service History** feature is enabled.	No

The following table lists and describes the attributes which appear in the adjustment cancellation response in the JSON format:

Attribute Name	Attribute Description
cancelAdjResponse	Indicates whether the adjustment cancellation request is successfully processed or not.
messageText	Indicates the error that occurred while canceling the adjustment. **Note:** This attribute appears when the value of the **statusCd** attribute is set to **Fail**.
cancelReasonDesc	Indicates the reason why the adjustment is canceled.
adjustmentStatus	Indicates the status of the adjustment. The valid value is: • Canceled
statusCd	Indicates whether the adjustment cancellation request is successfully processed or not. The valid values are: • Success • Fail

Related Topics

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
Sample Adjustment Cancellation Request and Response in the JSON Format	Sample Adjustment Cancellation Request and Response in the JSON Format on page 505

Sample Adjustment Cancellation Request and Response in the XML Format

Using the below example, you can cancel an adjustment which has incorrect data.

Sample Adjustment Cancellation Request in the XML Format

```
<C1-CancelAdjustment>
    <cancelAdjRequest>
        <adjustmentId>567845345789</adjustmentId>
        <reason>INCDT</reason>
        <sourceSystem>ES01</sourceSystem>
        <externalTransactionId>567845</externalTransactionId>
        <externalSourceId>2341</externalSourceId>
    </cancelAdjRequest>
</C1-CancelAdjustment>
```

Sample Adjustment Cancellation Response in the XML Format

```
<C1-CancelAdjustment>
    <cancelAdjRequest>
        <adjustmentId>567845345789</adjustmentId>
        <reason>INCDT</reason>
        <sourceSystem>ES01</sourceSystem>
        <externalTransactionId>567845</externalTransactionId>
        <externalSourceId>2341</externalSourceId>
    </cancelAdjRequest>
    <cancelAdjResponse>
        <cancelReasonDesc>Incorrect Data (INCDT)</cancelReasonDesc>
        <adjustmentStatus>Canceled</adjustmentStatus>
        <statusCd>Success</statusCd>
    </cancelAdjResponse>
</C1-CancelAdjustment>
```

Note: We have given the above sample to illustrate how the tags should be nested while sending an adjustment cancellation request from the external system. However, you should not seamlessly use this sample in your environment because the required pre-requisite data may not be available in your environment. In such case, the system will give erroneous results.

Related Topics

For more information on...	See...
Adjustment Cancellation Request and Response - Tags in XML Format	Adjustment Cancellation Request and Response - Tags in XML Format on page 501

Sample Adjustment Cancellation Request and Response in the JSON Format

Using the below example, you can cancel an adjustment which has incorrect data.

Sample Adjustment Cancellation Request in the JSON Format

```
{
    -"C1-CancelAdjustment": {
        -"cancelAdjRequest": {
            -"adjustmentId": -"333337888889",
```

Copyright © 2024, Oracle and/or its affiliates.

```
            -"reason": -"INNDT",
            -"sourceSystem": -"ES01",
            -"externalTransactionId": -"675688",
            -"externalSourceId": -"2341"
        -}
    -}
}
```

Sample Adjustment Cancellation Response in the JSON Format

```
{
    -"C1-CancelAdjustment": {
        -"cancelAdjRequest": {
            -"adjustmentId": -"333337888889",
            -"reason": -"INNDT",
            -"sourceSystem": -"ES01",
            -"externalTransactionId": -"675688",
            -"externalSourceId": -"2341"
        -}
        -"cancelAdjResponse": {
            -"messageText": -"Adjustment Cancel Reason is invalid.",
            -"cancelReasonDesc": -"",
            -"adjustmentStatus": -"",
            -"statusCd": -"Fail"
        -}
    -}
}
```

Note: We have given the above sample to illustrate how the attribute value pairs should be structured while sending an adjustment cancellation request from the external system. However, you should not seamlessly use this sample in your environment because the required pre-requisite data may not be available in your environment. In such case, the system will give erroneous results.

Related Topics

For more information on...	See...
Adjustment Cancellation Request and Response - Attribute/Value Pairs in JSON Format	Adjustment Cancellation Request and Response - Attribute/Value Pairs in JSON Format on page 503

C1-GetAdjustmentHistory Business Service

The **View Account Adjustments (C1-GetAdjustmentHistory)** business service enables you to view the adjustments of an account. You can call this business service from an external system through an inbound web service. If you want to send the adjustment view request in the XML and/or JSON format, you need to ensure that you create a REST based inbound web service in Oracle Revenue Management and Billing. However, if you want to send the adjustment view request in the XML format, you can create a SOAP or REST based inbound web service in Oracle Revenue Management and Billing.

While sending the adjustment view request, you need to specify either of the following:

- Account ID
- Account Identifier Type, Account Identifier, and Division

You can specify various criteria (such as, bill ID, adjustment type, contract type, adjustment status, adjustment creation date range) along with the account details to filter the list of adjustments. You can specify the maximum number of records that can be retrieved through the **C1-GetAdjustmentHistory** business service in the **Set Limit for IWS Retrieving Records** option type of the **C1-RECLMT** feature configuration. Accordingly, the system lists the number of adjustments in the adjustment view response. For example, if the value in the **Set Limit for IWS Retrieving Records**

Copyright © 2024, Oracle and/or its affiliates.

option type is set to 100 and the result set contains 125 adjustments, the system will list only 100 adjustments of the account from the result set in the descending order of the creation date. For more information about the **C1-RECLMT** feature configuration, see Setting the C1-RECLMT Feature Configuration on page 1096.

The system enables you to track the adjustment view response sent to the external system using the **Inbound Web Service History** feature.

For more information, see the **Inbound Web Service History** section in the *Oracle Revenue Management and Billing Administrative Guide*.

Related Topics

For more information on...	See...
Adjustment View Request in the XML Format	Adjustment View Request in the XML Format on page 507
Adjustment View Response in the XML Format	Adjustment View Response in the XML Format on page 508
Adjustment View Request in the JSON Format	Adjustment View Request in the JSON Format on page 509
Adjustment View Response in the JSON Format	Adjustment View Response in the JSON Format on page 509
Adjustment View Request and Response - Tags in XML Format	Adjustment View Request and Response - Tags in XML Format on page 510
Adjustment View Request and Response - Attribute/Value Pairs in JSON Format	Adjustment View Request and Response - Attribute/Value Pairs in JSON Format on page 513
Sample Adjustment View Request and Response in the XML Format	Sample Adjustment View Request and Response in the XML Format on page 516
Sample Adjustment View Request and Response in the JSON Format	Sample Adjustment View Request and Response in the JSON Format on page 518

Adjustment View Request in the XML Format

The following code snippet illustrates how the tags should be nested while receiving the adjustment view request in the XML format:

```
<C1-GetAdjustmentHistory>
    <getAdjRequest>
        <sourceSystem></sourceSystem>
        <externalSourceId></externalSourceId>
        <c1ExternalTransactionId></c1ExternalTransactionId>
        <billId></billId>
        <adjustmentType></adjustmentType>
        <saType></saType>
        <accountId></accountId>
        <division></division>
        <accountNumber></accountNumber>
        <accountIdentifierType></accountIdentifierType>
        <adjustmentStatus></adjustmentStatus>
        <adjustmentFromDate></adjustmentFromDate>
        <adjustmentToDate></adjustmentToDate>
    </getAdjRequest>
</C1-GetAdjustmentHistory>
```

Related Topics

For more information on...	See...
Adjustment View Response in the XML Format	Adjustment View Response in the XML Format on page 508

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
Adjustment View Request and Response - Tags in XML Format	Adjustment View Request and Response - Tags in XML Format on page 510
Sample Adjustment View Request and Response in the XML Format	Sample Adjustment View Request and Response in the XML Format on page 516

Adjustment View Response in the XML Format

The following code snippet illustrates how the tags are nested while sending the adjustment view response in the XML format:

```
<C1-GetAdjustmentHistory>
    <getAdjRequest>
        <sourceSystem></sourceSystem>
        <externalSourceId></externalSourceId>
        <c1ExternalTransactionId></c1ExternalTransactionId>
        <billId></billId>
        <adjustmentType></adjustmentType>
        <saType></saType>
        <accountId></accountId>
        <division></division>
        <accountNumber></accountNumber>
        <accountIdentifierType></accountIdentifierType>
        <adjustmentStatus></adjustmentStatus>
        <adjustmentFromDate></adjustmentFromDate>
        <adjustmentToDate></adjustmentToDate>
    </getAdjRequest>
    <AdjustmentResponse>
        <statusCd></statusCd>
        <messageText></messageText>
        <adjustmentList>
            <adjustmentId></adjustmentId>
            <adjustmentTypeCd></adjustmentTypeCd>
            <adjustmentType></adjustmentType>
            <currency></currency>
            <currentAmount></currentAmount>
            <adjustmentStatus></adjustmentStatus>
            <creDt></creDt>
        </adjustmentList>
    </AdjustmentResponse>
</C1-GetAdjustmentHistory>
```

Related Topics

For more information on...	See...
Adjustment View Request in the XML Format	Adjustment View Request in the XML Format on page 507
Adjustment View Request and Response - Tags in XML Format	Adjustment View Request and Response - Tags in XML Format on page 510
Sample Adjustment View Request and Response in the XML Format	Sample Adjustment View Request and Response in the XML Format on page 516

Copyright © 2024, Oracle and/or its affiliates.

Adjustment View Request in the JSON Format

The following code snippet illustrates how the attribute/value pairs should be structured while receiving the adjustment view request in the JSON format:

```
{
    -"C1-GetAdjustmentHistory": {
        -"getAdjRequest": {
            -"sourceSystem": -"",
            -"externalSourceId": -"",
            -"c1ExternalTransactionId": -"",
            -"billId": -"",
            -"adjustmentType": -"",
            -"saType": -"",
            -"accountId": -"",
            -"division": -"",
            -"accountNumber": -"",
            -"accountIdentifierType": -"",
            -"adjustmentStatus": -"",
            -"adjustmentFromDate": -"",
            -"adjustmentToDate": -""
        -}
    -}
}
```

Related Topics

For more information on...	See...
Adjustment View Response in the JSON Format	Adjustment View Response in the JSON Format on page 509
Adjustment View Request and Response - Attribute/Value Pairs in JSON Format	Adjustment View Request and Response - Attribute/Value Pairs in JSON Format on page 513
Sample Adjustment View Request and Response in the JSON Format	Sample Adjustment View Request and Response in the JSON Format on page 518

Adjustment View Response in the JSON Format

The following code snippet illustrates how the attribute/value pairs are structured while sending the adjustment view response in the JSON format:

```
{
    -"C1-GetAdjustmentHistory": {
        -"getAdjRequest": {
            -"sourceSystem": -"",
            -"externalSourceId": -"",
            -"c1ExternalTransactionId": -"",
            -"billId": -"",
            -"adjustmentType": -"",
            -"saType": -"",
            -"accountId": -"",
            -"division": -"",
            -"accountNumber": -"",
            -"accountIdentifierType": -"",
            -"adjustmentStatus": -"",
            -"adjustmentFromDate": -"",
            -"adjustmentToDate": -""
        -}
        -"AdjustmentResponse": {
            -"statusCd": -"",
            -"messageText": -"",
```

Copyright © 2024, Oracle and/or its affiliates.

```
            -"adjustmentList": {
                -"adjustmentId": -"",
                -"adjustmentTypeCd": -"",
                -"adjustmentType": -"",
                -"currency": -"",
                -"currentAmount": -"",
                -"adjustmentStatus": -"",
                -"creDt": -""
            -}
        -}
    -}
}
```

Related Topics

For more information on...	See...
Adjustment View Request in the JSON Format	Adjustment View Request in the JSON Format on page 509
Adjustment View Request and Response - Attribute/Value Pairs in JSON Format	Adjustment View Request and Response - Attribute/Value Pairs in JSON Format on page 513
Sample Adjustment View Request and Response in the JSON Format	Sample Adjustment View Request and Response in the JSON Format on page 518

Adjustment View Request and Response - Tags in XML Format

Note: We recommend you to refer the Adjustment View Request in the XML Format on page 507 and Adjustment View Response in the XML Format on page 508 topics in parallel while understanding the below mentioned tags. This will help you to understand how the tags are nested in the XML format.

Before calling the **C1-GetAdjustmentHistory** business service through an inbound web service, you need to ensure that the adjustment view request contains the following tags:

Tag Name	Tag Description	Mandatory (Yes or No)
C1-GetAdjustmentHistory	Used to indicate that you want to invoke the **C1-GetAdjustmentHistory** business service.	Yes
getAdjRequest	Used to specify additional criteria along with the details of the account whose adjustments you want to view.	Yes
sourceSystem	Used to indicate the external system from which the adjustment view request is sent.	No

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Tag Description	Mandatory (Yes or No)
	Note: You must specify an external system which is already defined in the system. The external system is used when the **Inbound Web Service History** feature is enabled. The specified external system should match the external system which is associated with the outbound message type specified in the **C1-IWSHIST** algorithm. If the external system is not specified in the adjustment view request, the system, by default, stamps the external system associated with the outbound message type (specified in the **C1-IWSHIST** algorithm) corresponding to the record in the **C1_IWS_HIST** table. For more information, see the **Inbound Web Service History** section in the *Oracle Revenue Management and Billing Administrative Guide*.	
externalSourceId	Used to specify the external source system ID. **Note:** The external source system ID is used when the **Inbound Web Service History** feature is enabled.	No
c1ExternalTransactionId	Used to indicate the transaction in the external system through which the adjustment view request is sent. **Note:** The external transaction ID is used when the **Inbound Web Service History** feature is enabled.	No
billId	Used when you want to view adjustments which are created against a particular bill. **Note:** You must specify a bill which already exists in the system.	No
adjustmentType	Used when you want to view adjustments which are created using a particular adjustment type. **Note:** You must specify an adjustment type which is already defined in the system.	No
saType	Used when you want to view adjustments which are created against contracts of a particular contract type. **Note:** You must specify a contract type which is already defined in the system.	No
accountId	Used to indicate the account whose adjustments you want to view.	Yes (Conditional)

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Tag Description	Mandatory (Yes or No)
		Note: This data is required when the values are not specified in the **accountNumber**, **accountIdentifierType**, and **division** tags.
division	Used to indicate the division to which the account belongs. **Note:** You must specify a division which is already defined in the system.	Yes (Conditional) **Note:** This data is required when the value is not specified in the **accountId** tag.
accountNumber	Used to indicate the account whose adjustments you want to view.	Yes (Conditional) **Note:** This data is required when the value is not specified in the **accountId** tag.
accountIdentifierType	Used to specify the account identifier type. **Note:** You must specify an account identifier type which is already defined in the system.	Yes (Conditional) **Note:** This data is required when the value is not specified in the **accountId** tag.
adjustmentStatus	Used when you want to view adjustments with a particular status. **Note:** You must specify a status which is already defined in the **ADJ_STATUS_FLG** lookup field.	No
adjustmentFromDate	Used when you want to view adjustments which are created from a particular date onwards. **Note:** The date should be earlier than the date specified in the **adjustmentToDate** tag.	Yes (Conditional) **Note:** This data is required when the value is specified in the **adjustmentToDate** tag.
adjustmentToDate	Used when you want to view adjustments which are created till a particular date. **Note:** The date should be later than the date specified in the **adjustmentFromDate** tag.	Yes (Conditional) **Note:** This data is required when the value is specified in the **adjustmentFromDate** tag.

Note: The date range specified through the **adjustmentFromDate** and **adjustmentToDate** tags cannot exceed 365 days.

The following table lists and describes the tags which appear in the adjustment view response in the XML format:

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Tag Description
AdjustmentResponse	Indicates whether the adjustment view request is successfully processed or not.
statusCd	Indicates whether the adjustment view request is successfully processed or not. The valid values are: • Success • Fail
messageText	Indicates the error that occurred while retrieving the adjustments of the account. **Note:** This tag appears when the value in the **statusCd** tag is set to **Fail**.
adjustmentList	Displays the details of an adjustment.
adjustmentId	Displays the adjustment ID.
adjustmentTypeCd	Indicates the adjustment type using which the adjustment is created.
adjustmentType	Displays the description of the adjustment type.
currency	Indicates the currency in which the adjustment is created.
currentAmount	Displays the adjustment amount.
adjustmentStatus	Indicates the status of the adjustment. The valid values are: • Canceled • Error • Freezable • Frozen • Incomplete • New
creDt	Displays the date when the adjustment is created in the system.

Related Topics

For more information on...	See...
Sample Adjustment View Request and Response in the XML Format	Sample Adjustment View Request and Response in the XML Format on page 516

Adjustment View Request and Response - Attribute/Value Pairs in JSON Format

Note: We recommend you to refer the Adjustment View Request in the JSON Format on page 509 and Adjustment View Response in the JSON Format on page 509 topics in parallel while understanding the below mentioned attributes. This will help you to understand how the attribute/value pairs are structured in the JSON format.

Before calling the **C1-GetAdjustmentHistory** business service through an inbound web service, you need to ensure that the adjustment view request contains the following attributes:

Attribute Name	Attribute Description	Mandatory (Yes or No)
C1-GetAdjustmentHistory	Used to indicate that you want to invoke the **C1-GetAdjustmentHistory** business service.	Yes
getAdjRequest	Used to specify additional criteria along with the details of the account whose adjustments you want to view.	Yes

Copyright © 2024, Oracle and/or its affiliates.

Attribute Name	Attribute Description	Mandatory (Yes or No)
sourceSystem	Used to indicate the external system from which the adjustment view request is sent. **Note:** You must specify an external system which is already defined in the system. The external system is used when the **Inbound Web Service History** feature is enabled. The specified external system should match the external system which is associated with the outbound message type specified in the **C1-IWSHIST** algorithm. If the external system is not specified in the adjustment view request, the system, by default, stamps the external system associated with the outbound message type (specified in the **C1-IWSHIST** algorithm) corresponding to the record in the **C1_IWS_HIST** table. For more information, see the **Inbound Web Service History** section in the *Oracle Revenue Management and Billing Administrative Guide*.	No
externalSourceId	Used to specify the external source system ID. **Note:** The external source system ID is used when the **Inbound Web Service History** feature is enabled.	No
c1ExternalTransactionId	Used to indicate the transaction in the external system through which the adjustment view request is sent. **Note:** The external transaction ID is used when the **Inbound Web Service History** feature is enabled.	No
billId	Used when you want to view adjustments which are created against a particular bill. **Note:** You must specify a bill which already exists in the system.	No
adjustmentType	Used when you want to view adjustments which are created using a particular adjustment type. **Note:** You must specify an adjustment type which is already defined in the system.	No
saType	Used when you want to view adjustments which are created against contracts of a particular contract type. **Note:** You must specify a contract type which is already defined in the system.	No
accountId	Used to indicate the account whose adjustments you want to view.	Yes (Conditional)

Copyright © 2024, Oracle and/or its affiliates.

Attribute Name	Attribute Description	Mandatory (Yes or No)
		Note: This data is required when the values are not specified for the **accountNumber**, **accountIdentifierType**, and **division** attributes.
division	Used to indicate the division to which the account belongs. **Note:** You must specify a division which is already defined in the system.	Yes (Conditional) **Note:** This data is required when the value is not specified for the **accountId** attribute.
accountNumber	Used to indicate the account whose adjustments you want to view.	Yes (Conditional) **Note:** This data is required when the value is not specified for the **accountId** attribute.
accountIdentifierType	Used to specify the account identifier type. **Note:** You must specify an account identifier type which is already defined in the system.	Yes (Conditional) **Note:** This data is required when the value is not specified for the **accountId** attribute.
adjustmentStatus	Used when you want to view adjustments with a particular status. **Note:** You must specify a status which is already defined in the **ADJ_STATUS_FLG** lookup field.	No
adjustmentFromDate	Used when you want to view adjustments which are created from a particular date onwards. **Note:** The date should be earlier than the date specified for the **adjustmentToDate** attribute.	Yes (Conditional) **Note:** This data is required when the value is specified for the **adjustmentToDate** attribute.
adjustmentToDate	Used when you want to view adjustments which are created till a particular date. **Note:** The date should be later than the date specified for the **adjustmentFromDate** attribute.	Yes (Conditional) **Note:** This data is required when the value is specified for the **adjustmentFromDate** attribute.

Note: The date range specified through the **adjustmentFromDate** and **adjustmentToDate** attributes cannot exceed 365 days.

The following table lists and describes the attributes which appear in the adjustment view response in the JSON format:

Copyright © 2024, Oracle and/or its affiliates.

Attribute Name	Attribute Description
AdjustmentResponse	Indicates whether the adjustment view request is successfully processed or not.
statusCd	Indicates whether the adjustment view request is successfully processed or not. The valid values are: • Success • Fail
messageText	Indicates the error that occurred while retrieving the adjustments of the account. **Note:** This attribute appears when the value of the **statusCd** attribute is set to **Fail**.
adjustmentList	Displays the details of an adjustment.
adjustmentId	Displays the adjustment ID.
adjustmentTypeCd	Indicates the adjustment type using which the adjustment is created.
adjustmentType	Displays the description of the adjustment type.
currency	Indicates the currency in which the adjustment is created.
currentAmount	Displays the adjustment amount.
adjustmentStatus	Indicates the status of the adjustment. The valid values are: • Canceled • Error • Freezable • Frozen • Incomplete • New
creDt	Displays the date when the adjustment is created in the system.

Related Topics

For more information on...	See...
Sample Adjustment View Request and Response in the JSON Format	Sample Adjustment View Request and Response in the JSON Format on page 518

Sample Adjustment View Request and Response in the XML Format

Using the below example, you can view the adjustments of the **3246723446** account whose data is received from the **ES01** system and which are created using the **ADTY1** adjustment type between 2016-01-01 and 2016-06-30.

Sample Adjustment View Request in the XML Format

```
<C1-GetAdjustmentHistory>
    <getAdjRequest>
        <sourceSystem>ES01</sourceSystem>
        <externalSourceId></externalSourceId>
        <c1ExternalTransactionId></c1ExternalTransactionId>
        <billId></billId>
        <adjustmentType>ADTY1</adjustmentType>
        <saType></saType>
        <accountId>3246723446</accountId>
        <division></division>
        <accountNumber></accountNumber>
```

Copyright © 2024, Oracle and/or its affiliates.

```
        <accountIdentifierType></accountIdentifierType>
        <adjustmentStatus></adjustmentStatus>
        <adjustmentFromDate>2016-01-01</adjustmentFromDate>
        <adjustmentToDate>2016-06-30</adjustmentToDate>
    </getAdjRequest>
</C1-GetAdjustmentHistory>
```

Sample Adjustment View Response in the XML Format

```
<C1-GetAdjustmentHistory>
    <getAdjRequest>
        <sourceSystem>ES01</sourceSystem>
        <externalSourceId></externalSourceId>
        <c1ExternalTransactionId></c1ExternalTransactionId>
        <billId></billId>
        <adjustmentType>ADTY1</adjustmentType>
        <saType></saType>
        <accountId>3246723446</accountId>
        <division></division>
        <accountNumber></accountNumber>
        <accountIdentifierType></accountIdentifierType>
        <adjustmentStatus></adjustmentStatus>
        <adjustmentFromDate>2016-01-01</adjustmentFromDate>
        <adjustmentToDate>2016-06-30</adjustmentToDate>
    </getAdjRequest>
    <AdjustmentResponse>
        <statusCd>Success</statusCd>
        <adjustmentList>
            <adjustmentId>143489080564</adjustmentId>
            <adjustmentTypeCd>ADTY1</adjustmentTypeCd>
            <adjustmentType>ADJUSTMENT TYPE 001</adjustmentType>
            <currency>USD</currency>
            <currentAmount>12.00</currentAmount>
            <adjustmentStatus>Frozen</adjustmentStatus>
            <creDt>2016-06-28</creDt>
        </adjustmentList>
        <adjustmentList>
            <adjustmentId>135489080564</adjustmentId>
            <adjustmentTypeCd>ADTY1</adjustmentTypeCd>
            <adjustmentType>ADJUSTMENT TYPE 001</adjustmentType>
            <currency>USD</currency>
            <currentAmount>75.00</currentAmount>
            <adjustmentStatus>Freezable</adjustmentStatus>
            <creDt>2016-05-21</creDt>
        </adjustmentList>
        <adjustmentList>
            <adjustmentId>130489080564</adjustmentId>
            <adjustmentTypeCd>ADTY1</adjustmentTypeCd>
            <adjustmentType>ADJUSTMENT TYPE 001</adjustmentType>
            <currency>USD</currency>
            <currentAmount>50.00</currentAmount>
            <adjustmentStatus>Frozen</adjustmentStatus>
            <creDt>2016-04-10</creDt>
        </adjustmentList>
        <adjustmentList>
            <adjustmentId>128426800005</adjustmentId>
            <adjustmentTypeCd>ADTY1</adjustmentTypeCd>
            <adjustmentType>ADJUSTMENT TYPE 001</adjustmentType>
            <currency>USD</currency>
            <currentAmount>15.00</currentAmount>
            <adjustmentStatus>Freezable</adjustmentStatus>
            <creDt>2016-03-15</creDt>
        </adjustmentList>
```

Copyright © 2024, Oracle and/or its affiliates.

```
        <adjustmentList>
            <adjustmentId>127989656645</adjustmentId>
            <adjustmentTypeCd>ADTY1</adjustmentTypeCd>
            <adjustmentType>ADJUSTMENT TYPE 001</adjustmentType>
            <currency>USD</currency>
            <currentAmount>25.00</currentAmount>
            <adjustmentStatus>Frozen</adjustmentStatus>
            <creDt>2016-02-25</creDt>
        </adjustmentList>
        <adjustmentList>
            <adjustmentId>123489080564</adjustmentId>
            <adjustmentTypeCd>ADTY1</adjustmentTypeCd>
            <adjustmentType>ADJUSTMENT TYPE 001</adjustmentType>
            <currency>USD</currency>
            <currentAmount>10.00</currentAmount>
            <adjustmentStatus>Frozen</adjustmentStatus>
            <creDt>2016-01-28</creDt>
        </adjustmentList>
    </AdjustmentResponse>
</C1-GetAdjustmentHistory>
```

Note: We have given the above sample to illustrate how the tags should be nested while sending an adjustment view request from the external system. However, you should not seamlessly use this sample in your environment because the required pre-requisite data may not be available in your environment. In such case, the system will give erroneous results.

Related Topics

For more information on...	See...
Adjustment View Request and Response - Tags in XML Format	Adjustment View Request and Response - Tags in XML Format on page 510

Sample Adjustment View Request and Response in the JSON Format

Using the below example, you can view the adjustments of an account which are created against the **5672345000** bill.

Sample Adjustment View Request in the JSON Format

```
{
    -"C1-GetAdjustmentHistory": {
        -"getAdjRequest": {
            -"sourceSystem": -"",
            -"externalSourceId": -"",
            -"c1ExternalTransactionId": -"",
            -"billId": -"5672345000",
            -"adjustmentType": -"",
            -"saType": -"",
            -"accountId": -"",
            -"division": -"D0307",
            -"accountNumber": -"2345678900",
            -"accountIdentifierType": -"ACCTNUM"",
            -"adjustmentStatus": -"",
            -"adjustmentFromDate": -"",
            -"adjustmentToDate": -""
        -}
    -}
}
```

Sample Adjustment View Response in the JSON Format

```
{
```

Copyright © 2024, Oracle and/or its affiliates.

```
    -"C1-GetAdjustmentHistory": {
        -"getAdjRequest": {
            -"sourceSystem": -"",
            -"externalSourceId": -"",
            -"c1ExternalTransactionId": -"",
            -"billId": -"5672345000",
            -"adjustmentType": -"",
            -"saType": -"",
            -"accountId": -"",
            -"division": -"D0307",
            -"accountNumber": -"2345678900",
            -"accountIdentifierType": -"ACCTNUM"",
            -"adjustmentStatus": -"",
            -"adjustmentFromDate": -"",
            -"adjustmentToDate": -""
        -}
        -"AdjustmentResponse": {
            -"statusCd": -"Success",
            -"adjustmentList": {
                -"adjustmentId": -"656565656555",
                -"adjustmentTypeCd": -"ADTY1",
                -"adjustmentType": -"ADJUSTMENT TYPE 001",
                -"currency": -"USD",
                -"currentAmount": -"12.00",
                -"adjustmentStatus": -"Frozen",
                -"creDt": -"2020-01-28"
            -},
            -"adjustmentList": {
                -"adjustmentId": -"656565656543",
                -"adjustmentTypeCd": -"ADTY1",
                -"adjustmentType": -"ADJUSTMENT TYPE 001",
                -"currency": -"USD",
                -"currentAmount": -"10.00",
                -"adjustmentStatus": -"Frozen",
                -"creDt": -"2020-01-25"
            -},
            -"adjustmentList": {
                -"adjustmentId": -"656565656523",
                -"adjustmentTypeCd": -"ADTY2",
                -"adjustmentType": -"ADJUSTMENT TYPE 002",
                -"currency": -"USD",
                -"currentAmount": -"15.00",
                -"adjustmentStatus": -"Frozen",
                -"creDt": -"2020-01-21"
            -},
            -"adjustmentList": {
                -"adjustmentId": -"656565656519",
                -"adjustmentTypeCd": -"ADTY5",
                -"adjustmentType": -"ADJUSTMENT TYPE 005",
                -"currency": -"USD",
                -"currentAmount": -"13.00",
                -"adjustmentStatus": -"Frozen",
                -"creDt": -"2020-01-18"
            -},
            -"adjustmentList": {
                -"adjustmentId": -"656565656490",
                -"adjustmentTypeCd": -"ADTY4",
                -"adjustmentType": -"ADJUSTMENT TYPE 004",
                -"currency": -"USD",
                -"currentAmount": -"20.00",
                -"adjustmentStatus": -"Frozen",
                -"creDt": -"2020-01-14"
            -},
            -"adjustmentList": {
```


Copyright © 2024, Oracle and/or its affiliates.

```
                -"adjustmentId": -"656565656487",
                -"adjustmentTypeCd": -"ADTY3",
                -"adjustmentType": -"ADJUSTMENT TYPE 003",
                -"currency": -"USD",
                -"currentAmount": -"21.00",
                -"adjustmentStatus": -"Frozen",
                -"creDt": -"2020-01-13"
            -},
            -"adjustmentList": {
                -"adjustmentId": -"656565656456",
                -"adjustmentTypeCd": -"ADTY2",
                -"adjustmentType": -"ADJUSTMENT TYPE 002",
                -"currency": -"USD",
                -"currentAmount": -"10.00",
                -"adjustmentStatus": -"Frozen",
                -"creDt": -"2020-01-11"
            -},
            -"adjustmentList": {
                -"adjustmentId": -"656565656423",
                -"adjustmentTypeCd": -"ADTY5",
                -"adjustmentType": -"ADJUSTMENT TYPE 005",
                -"currency": -"USD",
                -"currentAmount": -"18.00",
                -"adjustmentStatus": -"Frozen",
                -"creDt": -"2020-01-07"
            -},
            -"adjustmentList": {
                -"adjustmentId": -"656565656422",
                -"adjustmentTypeCd": -"ADTY3",
                -"adjustmentType": -"ADJUSTMENT TYPE 003",
                -"currency": -"USD",
                -"currentAmount": -"11.00",
                -"adjustmentStatus": -"Frozen",
                -"creDt": -"2020-01-05"
            -}
        -}
    -}
}
```


Note: We have given the above sample to illustrate how the attribute value pairs should be structured while sending an adjustment view request from the external system. However, you should not seamlessly use this sample in your environment because the required pre-requisite data may not be available in your environment. In such case, the system will give erroneous results.

Related Topics

For more information on...	See...
Adjustment View Request and Response - Attribute/Value Pairs in JSON Format	Adjustment View Request and Response - Attribute/Value Pairs in JSON Format on page 513

Chapter 5

Credit & Collections

Topics:

- The Big Picture Of Credit & Collections
- Collection Process Maintenance
- Write-off Process Maintenance
- Writing Off Uncollectable Debt
- Collection Referral
- Payment Arrangements
- Promise To Pay

The system periodically monitors how much your customers owe to ensure they haven't violated your collection criteria. When a violation is detected, the system generates the appropriate events (e.g., letters, ToDo entries, write-off outstanding debt, etc.). This section describes how to manage your credit & collections processing.

Note: In the section, The Big Picture Of Credit & Collections (C&C), we describe how to set up the control tables that automate most of your credit & collections processing. If you set these tables up properly, the system will manage most of your collection and write-off processes; little or no user intervention should be necessary. You will only access the C&C pages when you need to override what the system does.

The Big Picture Of Credit & Collections

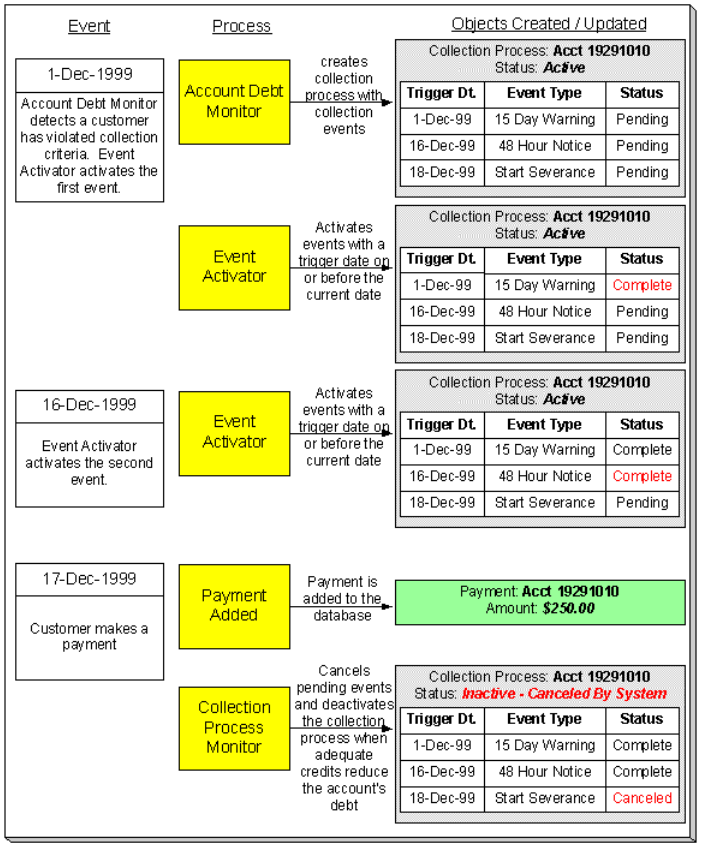
The topics in this section provide background information about a variety of credit & collections issues.

Automating Your C&C Activities

Fastpath: Refer to The Big Picture Of Credit & Collections (C&C) to understand how the system monitors overdue debt and initiates the appropriate collection and write-off events when a customer exceeds your debt tolerances.

The Lifecycle Of A Collection Process And Its Events

The following diagram illustrates the lifecycle of a typical collection process and its collection events:



The above diagram is referenced in the following topics.

What Is A Collection Process?

A collection process is a series of events (e.g., letters, ToDo entries) meant to encourage an account to pay its delinquent debt. Linked to the collection process are the specific contracts that contributed to the delinquent debt.

How Are Collection Processes Created?

A collection process and its events may be created as follows:

Copyright © 2024, Oracle and/or its affiliates.

- The Account Debt Monitor is a background process that creates a collection process when an account violates your collection criteria. The collection process has one or more collection event(s). The collection process template associated with the collection process controls the number and type of events.
- A user may create an ad hoc collection process at will. The collection process template controls the number and type of events linked to the collection process.

In the above diagram, the event on 1-Dec-99 shows the creation of a collection process with three collection events. Notice that the events all exist in the `Pending` state. The events will exist in this state until their trigger date is on or before the current date.

Fastpath: Refer to How Does The Account Debt Monitor Work? to understand how the system monitors overdue debt and initiates the appropriate collection process when a customer exceeds your debt tolerances. Refer to The Big Picture Of Collection Events for more information about a collection process' events.

Fastpath: When you enable the Control Central alert algorithm, C1-COLLPR-AC, an alert displays when an account has an active collection process. This algorithm is plugged-in on the installation record.

How Are Collection Events Completed?

The event activator runs periodically (at least daily). It looks for collection events with a trigger date on or before the current date. For each such event, the system executes its activity and then completes it.

In the above diagram, the event activator activates pending events on 1-Dec-99 and 16-Dec-99. Notice that the events are moved to the `Complete` state.

Fastpath: Refer to The Big Picture Of Collection Events for more information about the various types of collection events and what happens when they are completed.

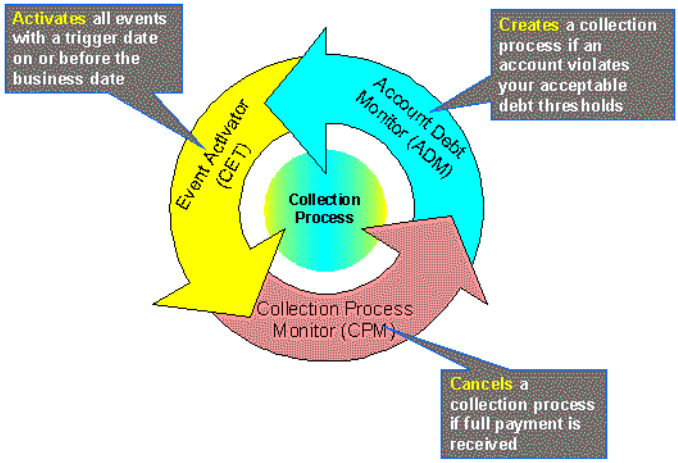
How Are Collection Processes (and their Events) Canceled?

Fastpath: Refer to How Are Collection Processes Cancelled for a description of how and when collection processes are automatically cancelled by the system.

Besides the automated cancellation process, a user may cancel a collection process at will. Refer to How To Cancel A Collection Process for more information.

The Collection Process Background Processes

There are 3 background processes that automate the creation, activation, and cancellation of collection processes:



Copyright © 2024, Oracle and/or its affiliates.

Account Debt Monitor

The Account Debt Monitor (ADM) creates a collection process when an account violates your collection criteria. The collection process has one or more collection event(s). The number and type of events is defined on the collection process's template.

> **Note: Heads up.** There are actually two background processes involved with the monitoring of overdue debt. These processes are referred to using the batch process ID's of `ADM` and `ADM2`.

In the above diagram, the ADM creates a collection process (with 3 events) on 1-Dec-99. Notice that the events all exist in the `Pending` state. The events will exist in this state until their trigger date is on or before the current date.

> **Fastpath:** Refer to How Does The Account Debt Monitor Work for more information.

Collection Event Activator

The collection event activator runs periodically (at least daily). It looks for collection events with a trigger date on or before the current date. For each such event, the system executes its activity and then completes it.

> **Note:** This batch process is referred to using the batch ID `CET`.

In the above diagram, the collection event activator activates pending events on 1-Dec-99 and 16-Dec-99. Notice that the events are moved to the `Complete` state after they are activated.

Collection Process Monitor

The Collection Process Monitor runs periodically (at least daily). This process reviews a collection process whenever one of its contracts' debt is reduced.

> **Note:** This batch process is referred to using the batch ID `CPM`.

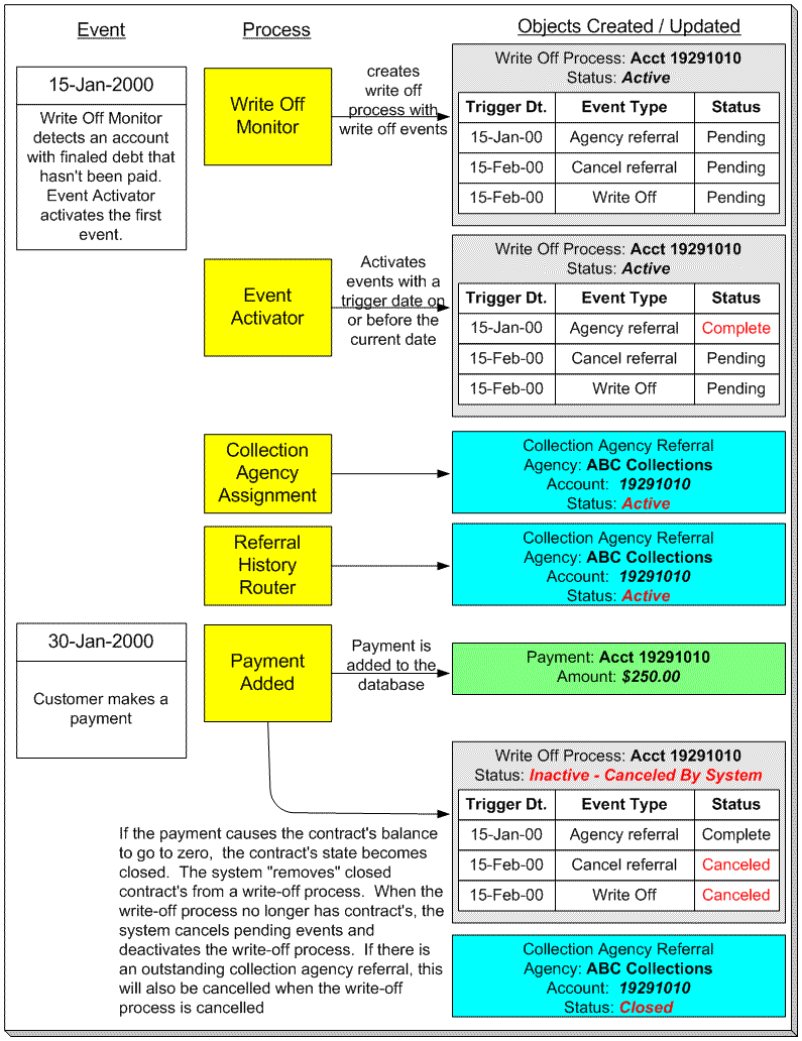
At review time, the Collection Process Monitor determines if it can cancel the collection process (because the customer's debt is no longer in a bad state). Refer to How Does The Collection Process Monitor Work for details describing how this process works.

In the above diagram, the collection process monitor cancels the process on 17-Dec-99. Notice that the events are moved to the `Cancelled` state and the process becomes `Inactive`.

The Lifecycle Of A Write Off Process And Its Events

The following diagram illustrates the lifecycle of a typical write-off process and its write-off events:

Copyright © 2024, Oracle and/or its affiliates.



The above diagram is referenced in the following topics.

What Is A Write-off Process?

A write-off process is a series of events (e.g., letters, collection agency referrals) meant to encourage an account to pay its delinquent debt. Linked to the write-off process are the specific contracts that contributed to the delinquent debt.

How Are Write-off Processes Created?

A write-off process and its events may be created as follows:

- The Write Off Monitor is a background process (referred to by the batch ID `WPM`) that creates a write-off process when an account has finaled, unpaid contracts. The write-off process has one or more write-off event(s). The write-off process template associated with the write-off process controls the number and type of events.
- A user may create an ad hoc write-off process at will. The write-off process template controls the number and type of events linked to the write-off process.

In the above diagram, the event on 15-Jan-00 shows the creation of a write-off process with three write-off events. Notice that the events all exist in the `Pending` state. The events will exist in this state until their trigger date is on or before the current date.

Fastpath: Refer to How Does The Write-Off Monitor Work? to understand how the system monitors overdue debt and initiates the appropriate write-off process when a customer has unpaid, finaled contracts. Refer to The Big Picture Of Write-off Events for more information about a write-off process' events.

Copyright © 2024, Oracle and/or its affiliates.

How Are Write-off Events Completed?

The event activator runs periodically (at least daily). It looks for write-off events with a trigger date on or before the current date. For each such event, the system executes its activity and then completes it.

In the above diagram, the event activator activates a pending event on 15-Jan-00. Notice that the events are moved to the `Complete` state.

Fastpath: Refer to The Big Picture Of Write-off Events for more information about the various types of write-off events and what happens when they are completed.

How Are Write-off Processes (and their Events) Canceled?

When a stopped contract's balance is reduced to zero (i.e., it closes), the system checks if there is an ongoing write-off process associated with the contract. If so:

- The system "removes" the contract from the write-off process.
- When the last contract is removed from a write-off process, the system cancels all pending write-off events and deactivates the write-off process.
- If the canceled process has collection agency referrals, the system cancels the referrals.

Note: Real time cancellation. Unlike collection processes, the system cancels write-off processes real time (i.e., there is no equivalent of the collection process monitor for write-off processes). This means that if a customer (or collection agency) pays for a contract that is linked to write-off process, the system will cancel the process and its collection agency referrals (if any) the moment the payment is entered.

Fastpath: Refer to Setting Up Write Off Process Templates for more information about write off process templates.

Besides the automated cancellation process, a user may cancel a write-off process at will.

What Happens If The Debt Is Not Paid?

If a payment (or some other type of credit) is not received by the trigger date of the last event in the process, the debt will be written off (assuming you designed your write off process template to do such).

Note: The last write-off event should cause debt to be written off. Debt will only be written off if the last write-off event is the type that writes off debt. If the write-off process template contains no such event, nothing further will transpire until the next time the write-off monitor runs (when it will create another write-off process for the unpaid, finaled debt).

Fastpath: Refer to How Is Debt Financially Written-Off? for a description of exactly how debt is written off.

How Is Debt Referred To Collection Agencies

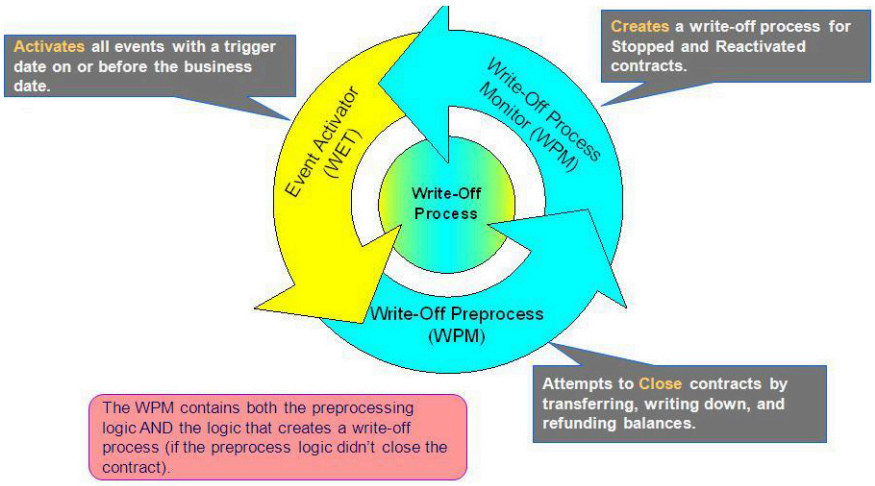
Fastpath: Refer to How Do Collection Agency Referrals Work? for more information about how the system refers debt to a collection agency.

The Write-Off Process Background Processes

There are 2 background processes that automate the creation, activation, and cancellation of write-off processes:

Note: If necessary, you can write-off debt outside of the automated write-off process by simply transferring bad debt to a write-off contract using the Write Off transaction.

Copyright © 2024, Oracle and/or its affiliates.



Write-Off Monitor

The Write-Off Monitor is a background process that has two functions:

- It attempts to close Stopped and Reactivated contracts using several methods as described under Attempt To Close The Contract Before Creating A Write Off Process.
- If it is unsuccessful at closing a contract, it creates a write-off process. The write-off process has one or more write-off event(s). The write-off process template associated with the write-off process controls the number and type of events.

Note: This batch process is referred to using the batch ID `WPM`.

In the above diagram, the event on 15-Jan-2000 shows the creation of a write-off process with three write-off events. Notice that the events all exist in the `Pending` state. The events will exist in this state until their trigger date is on or before the current date.

Write-Off Event Activator

The write-off event activator runs periodically (at least daily). It looks for write-off events with a trigger date on or before the current date. For each such event, the system executes its activity and then completes it.

Note: This batch process is referred to using the batch ID `WET`.

In the above diagram, the write-off event activator activates a pending event on 15-Jan-2000. Notice that the event is moved to the `Complete` state.

Collection Process Maintenance

A collection process is a series of events (e.g., letters, ToDo entries) meant to encourage an account to pay its delinquent debt. Linked to the collection process are the specific contracts that contributed to the delinquent debt. The topics in this section describe the pages on which collection process-related information is maintained.

Fastpath: For more information about collection processes, refer to The Lifecycle Of A Collection Process And Its Events.

Collection Process - Main Information

The Main page contains core collection process information. Open this page using **Delinquency Management, Collection Process**.

Copyright © 2024, Oracle and/or its affiliates.

Fastpath: The **Description of Page** section below describes the fields on this page. Refer to How To Perform Common Collection Process Functions for more instructions describing how to use this transaction.

Description of Page

Collection Process is a concatenation of summary information about this collection process. It is composed of the name of the main customer on the account, the collection status, the reason code for the collection status, the collection class control name, and the creation date of the process.

Collection Process ID is the unique system-generated identifier for this collection process.

Account ID is the system-generated account number. The name of the main customer on the account and its customer class are displayed next to the ID number.

Collection Status defines the state of the collection process. The following values may exist:

`Active` The system creates a collection process in the `Active` state. A collection process remains in this state until the last event linked to the process is complete OR the customer pays the collection amount on the contracts linked to the process OR you want to stop the process for whatever reason.

`Inactive` A collection process becomes `Inactive` when the last event linked to the process is completed OR the customer pays the collection amount on all contracts linked to the process OR you want to stop the process for whatever reason. The **Reason** field tells you how a collection process became inactive.

Fastpath: To completely understand the significance of a collection process' status, refer to The Lifecycle Of A Collection Process And Its Events.

Reason defines how a collection process became `Inactive` or why a collection process is `Active`. The following values may exist:

`Canceled by User` This reason should be used when a user manually inactivates a collection process (for whatever reason).

`Canceled by System` This reason is populated by the system when it cancels a collection process. The system automatically cancels a collection process when sufficient credits have been posted to the process' contracts since the process was started.

`Events Pending` This reason is populated by the system when a process has pending events.

`Completed` This reason is populated by the system when it completes the last event linked to the collection process.

Collection Class Control defines the debt class (i.e., the type of debt) associated with the collection process. This is important as it controls which contracts can be linked to the collection process (every contract is associated with a specific debt class). This field is gray when there are contracts linked to the collection process.

Note: Multiple collection processes may be linked to an account. It's important to be aware that if an account's contracts reference multiple debt classes, a collection process must be created for each debt class that you want to collect.

Note: One collection process per debt class. A given debt class for an account may only have one ongoing collection process at any point in time.

Beneath **Collection Class Control** is displayed a message showing why the system generated the collection process. This information is only displayed for collection processes created by the Account Debt Monitor. If you have plugged in a `Collection Process Additional Information` algorithm on the Installation table, the results of the algorithm, if any, is displayed on a separate line.

Fastpath: For more information, refer to How Does The Account Debt Monitor Work?.

Collection Process Template defines the collection events in the process. You can override these events on the **Events** page.

Copyright © 2024, Oracle and/or its affiliates.

Note: Changing the template. If you change the template when all events are `pending`, the system will remove the `pending` events and replace them with the new template's events.

Fastpath: For more information, refer to Setting Up Collection Process Templates.

Creation Date/Time defines the start date/time of the collection process. This is important as it affects the trigger dates of the process' collection events. This field becomes protected after the collection process exists on the database. If you need to change the trigger date on the events you can navigate to the **Events** tab and change the dates manually. Alternatively, you can perform the following "trick":

- Change the **Collection Process Template** to something (anything). Doing this causes the **Create Date/Time** to become unprotected.
- Specify the date of the first collection event in the newly unprotected **Create Date/Time**.
- Change the **Collection Process Template** back to the desired value. When you change the template, the system removes the events and creates new ones using the **Create Date/Time**.

The collection process cancellation logic will cancel a collection process when its account's debt no longer exceeds a given amount (you define this threshold amount when you set up the system). Rather than compare an account's total debt to the threshold amount, the cancellation logic compares the account's debt that is older than `X` days to the threshold amount. `X` is the date specified in **Collection Amount Base Date**. You can have the system calculate this date for you by entering the age of the arrears in the field embedded in **Calculate Date Using _ Days In Arrears**.

Fastpath: For more information, refer to How Are Collection Processes (and their Events) Canceled?

Enter any **Comments** about the collection process.

The tree at the bottom of the page shows a variety of information about the **account** and the collection process including:

- The contracts linked to the collection process.
- The events linked to the collection process.
- The amount of debt currently associated with the account's debt classes.
- The minimum value payment to the collection process debt class that will cause cancellation of the collection process.
- Whether a promise to pay exists for the account, or existed at the time the collection process as created.
- The balance for any contracts linked to the collection process in Inactive status but that have credit balances contributing to the overall debt class arrears for the account.
- A summary of other active collection activities (i.e., collection and write-off processes) associated with the **account**.

Collection Process - Contracts

The Contract page contains the contracts whose debt is being collected by the collection process. Open this page using **Delinquency Management, Collection Process, Contracts**.

Fastpath: The **Description of Page** section below describes the fields on this page. Refer to How To Perform Common Collection Process Functions for more instructions describing how to use this transaction.

Description of Page

The top portion of every page in this page contains the following information:

Collection Process is a concatenation of summary information about this collection process. It is composed of the name of the main customer on the account, the collection status, the reason code for the collection status, the collection class control name, and the creation date of the process.

ID is the unique system-generated identifier for this collection process.

The **Collection Contracts** scroll contains the contract being collected under this collection process. To modify a contract linked to the collection process, simply move to a field and change its value. To add a new contract, press the + button to insert a row, then fill in the information for each field. The following fields display:

Copyright © 2024, Oracle and/or its affiliates.

Contract ID identifies the contract being collected. Information about the contract is displayed beneath the contract ID.

Status defines the state of the contract in respect of the collection process. The following values may exist:

`Active` An `Active` contract is one currently being collected by the collection process. The contract will remain in this state until the last event linked to the process is complete OR the customer pays the collection amount on the contract OR you want to remove the contract from the collection process for whatever reason.

`Inactive` A contract becomes `Inactive` within the collection process when the last event linked to the process is completed OR the customer pays the collection amount on the contract OR you want to remove the contract from the collection process for whatever reason

The **Collection Amount** is the amount of debt that initially triggered the collection process. This will be zero for collection processes that were created manually.

The **Amount Still Owing** is the amount of debt older than the **Collection Amount Base Date** defined on the Main page.

The **Credit Amount** is displayed if the Amount Still Owing is less that zero.

The **Contract Balances** area displays how much is owed for the contract. The **Days Old** and **Arrears Amount** information shows how old the current balance is. This information will be blank if the current balance is a credit.

Note: Information about debt balances by age is not applicable to open-item accounts.

Current Balance indicates the amount the customer currently owes for the service. **Payoff Balance** indicates the amount the customer would owe if they wanted to close the contract. This value is only displayed when it differs from the current balance.

Fastpath: Refer to Current Amount versus Payoff Amount for more information.

Collection Process - Events

The events page contains the activities that will be performed to persuade the customer to pay the outstanding debt. Open this page using **Delinquency Management, Collection Process, Events**.

Fastpath: The **Description of Page** section below describes the fields on this page. Refer to How To Perform Common Collection Process Functions for more instructions describing how to use this transaction.

Note: We strongly recommend that you understand the information described in How Are Collection Events Completed? before using this page.

The number and type of events associated with a collection process are defined by the process's collection process template (which is defined on the first page). Refer to Setting Up Collection Process Templates for more information.

Description of Page

The top portion of every page in this page contains the following information:

Collection Process is a concatenation of summary information about this collection process. It is composed of the name of the main customer on the account, the collection status, the reason code for the collection status, the collection class control name, and the creation date of the process.

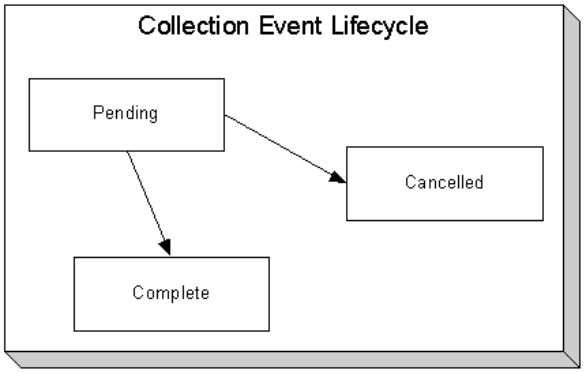
ID is the unique system-generated identifier for this collection process.

The **Collection Events** scroll contains the collection events that comprise the collection process. Only one event is displayed at a time. The inner scroll controls the display of the customer contacts that were created by the system when a particular collection event was triggered.

Event Sequence is the unique identifier of the event.

Collection Event Status defines the state of the event. The following diagram shows the possible lifecycle of a collection event:

Copyright © 2024, Oracle and/or its affiliates.



Collection events are initially created in the `pending` state.

When the system sees a `pending` event with a **Trigger Date** on or before the current date, the system executes the event's activity and `completes` the event.

You can cancel an individual collection event by setting the status to `cancelled`.

The **Collection Event Type** defines the event's activity (e.g., Email sent, a ToDo entry is generated, a letter is sent).

Fastpath: For more information, refer to Setting Up Collection Process Templates.

The **Trigger Date** defines the date when the system completes (i.e., executes) the event.

Fastpath: For more information about a collection event's trigger date, see Collection Event Trigger Date.

If the status of the event is `complete`, the **Completion Date** is displayed with the date on which the system completes the event. If the status of the event is `cancelled`, the **Canceled Date** is displayed with the date on which the event was system canceled.

The **Customer Contact** information contains information when the event template causes a letter to be generated (because letters are created by customer contacts).

How To Perform Common Collection Process Functions

The topics in this section describe how to perform common collection process maintenance functions. Refer to The Lifecycle Of A Collection Process for more information about collection processes. Refer to The Big Picture Of Collection Processing for high-level information about collection processing.

Related Topics:

- How To Create A Collection Process on page 531
- How To Change Collection Events on page 532
- How To Cancel A Collection Process on page 532

Parent Topic: Collection Process Maintenance on page 527

How To Create A Collection Process

99.9% of all collection processes are created by the Account Debt Monitor and require no human intervention before they are executed (refer to How Are Collection Processes Created for information about how the account debt monitor creates collection processes). The other 0.1% are created by users on-line / real time. The following points describe how to create the 0.1%.

- Use **Control Central** to choose the account that requires a new collection process.
- After the account is populated on Control Central, choose the **Collection Process +** option on the account context menu to transfer to the collection process transaction in add mode for the account.

Copyright © 2024, Oracle and/or its affiliates.

- After the Collection Process - Main page appears, specify the appropriate **Collection Class Control**. **Collection Class Control** defines the debt class (i.e., the type of debt) associated with the collection process. This is important as it controls which contracts can be linked to the collection process (every contract is associated with a specific debt class). It's important to be aware that if an account's contracts reference multiple debt classes, a collection process must be created for each debt class that you want to collect. Also be aware that a given debt class for an account may only have one ongoing collection process at any point in time.
- Choose a **Collection Process Template**. **Collection Process Template** defines the collection events in the process. You can override these events on the Events page. For more information, refer to Setting Up Collection Process Templates.
- Use **Collection Amount Base Date** to define the arrears cutoff date that the system uses when it "removes" a contract from a collection process. The system removes a contract from a collection process when its debt older than `X` is less than or equal to zero. `X` is the date specified in **Collection Amount Base Date**. You can have the system calculate this date for you by entering the age of the arrears in the field embedded in **Calculate Date Using _ Days In Arrears**.
- Navigate to the **Contracts** page to define the specific contract's whose debt is covered by this collection process. You must define at least one contract.
- Save the collection process.

How To Change Collection Events

When a collection process is first created, it has one or more collection events. The events are the activities that will be performed to persuade the customer to pay the outstanding debt.

The number and type of events that are created when a collection process is initiated are defined on the collection process's collection process template. The following points describe how to add / change / delete events on a collection process if the defaulted events are not satisfactory.

- Use **Control Central** to choose the account with the collection process whose events need to be changed.
- After the account is populated on Control Central, choose the **Collection Process** option on the account context menu to transfer to the collection process transaction in update mode for the account.
- To add a new event, transfer to the **Events** tab and press the + button in the **Collection Events** scroll to add a new event. At this point, the event has not been added to the database; rather, it just exists in memory. Before you add the event to the database, you must specify the following information:
  - Choose an **Event Sequence** so that the new event will be positioned properly in respect of the other events.
  - Choose a **Collection Event Status** of `Pending`.
  - Choose the desired **Collection Event Type**.
  - Use **Trigger Date** to define the date on which the event should be activated (i.e., completed).
  - To delete an existing event, transfer to the **Events** tab and use the scroll arrows to toggle to the desired collection event. When the desired collection event appears, press the - button to remove the event. At this point, the event has not been removed from the database; rather, it's been removed in memory.
  - To change an existing event, transfer to the **Events** tab and use the scroll arrows to toggle to the desired collection event. When the desired collection event appears, make the desired changes.
  - After all desired changes have been made, save the collection process.

How To Cancel A Collection Process

The Collection Process Monitor will automatically "remove" contracts from a collection process when their debt has been reduced sufficiently. When all contracts are "removed" from a collection process, the collection process is canceled. The following points describe how to manually cancel a collection process.

- Use **Control Central** to choose the account with the collection process to be cancelled.
- After the account is populated on Control Central, choose the **Collection Process** option on the account context menu to transfer to the collection process transaction in update mode for the account. A list of all collection processes associated with the account appears. If only one collection process exists, it is automatically selected for you.
- On the **Main** tab, click Cancel.
- When the warning dialog appears, click OK to continue. The collection process and its pending events are canceled.

Copyright © 2024, Oracle and/or its affiliates.

Write-off Process Maintenance

A write-off process is a series of events (e.g., letters, collection agency referrals) meant to encourage an account to pay its unpaid, finaled debt. Linked to the write-off process are the specific contracts that contributed to the delinquent debt. The topics in this section describe the pages on which write-off process-related information is maintained.

Fastpath: For more information about write-off processes, refer to The Lifecycle Of A Write Off Process And Its Events.

Write-off Process - Main Information

The Main page contains core write-off process information. Open this page using **Accounting and Receivable, Write Off Process**.

Fastpath: The **Description of Page** section below describes the fields on this page. Refer to How To Perform Common Write-Off Process Functions for more instructions describing how to use this transaction.

Description of Page

The top portion of every page contains the following information:

Write Off Process is a concatenation of summary information about this write-off process. It is composed of the name of the main customer on the account, the write-off status, the reason code for the write-off status, the write-off class control name, and the creation date of the process.

Write Off Process ID is the unique system-generated identifier for this write-off process.

The remaining fields are maintained on the main write-off process page.

Account ID is the system-generated account number. The name of the main customer on the account and its customer class are displayed adjacent.

Write-off Status defines the state of the write-off process. The following values may exist:

`Active` The system creates a write-off process in the `Active` state. A write-off process remains in this state until the last event linked to the process is complete OR the finaled debt is paid off OR you want to stop the process for whatever reason.

`Inactive` A write-off process becomes `Inactive` when the last event linked to the process is completed OR the finaled debt is paid off OR you want to stop the process for whatever reason. The **Reason** field tells you how a write-off process became inactive.

Fastpath: To completely understand the significance of a write-off process' status, refer to The Lifecycle Of A Write Off Process And Its Events.

Reason defines how a write-off process became `Inactive` or why a write-off process is `Active`. The following values may exist:

`Canceled by User` This reason should be used when a user manually inactivates a write-off process (for whatever reason).

`Canceled by System` This reason is populated by the system when it cancels a write-off process. The system automatically cancels a write-off process when all finaled debt associated with the process' contracts is paid.

`Events Pending` This reason is populated by the system when a process has pending events.

`Completed` This reason is populated by the system when it completes the last event linked to the write-off process.

Copyright © 2024, Oracle and/or its affiliates.

Write Off Control defines the write-off debt class (i.e., the type of debt) associated with the write-off process. This is important as it controls which contracts can be linked to the write-off process (every contract is associated with a specific write-off debt class). This field is gray when there are contracts linked to the write-off process.

Note: Multiple write-off processes may be linked to an account. It's important to be aware that if an account's contracts reference multiple write-off debt classes, a write-off process is created for each write-off debt class that you want to write-off.

Note: Multiple write-off processes per write-off debt class. A given write-off debt class for an account may have multiple ongoing write-off processes at any point in time. However, a given contract may only be on one active write-off process at any point in time.

Write Off Process Template defines the write-off events in the process. You can override these events on the **Events** page.

Note: Changing the template. If you change the template when all events are `pending`, the system will remove the `pending` events and replace them with the new template's events.

Fastpath: For more information, refer to Setting Up Write Off Process Templates.

Create Date/Time defines the start date/time of the write-off process. This is important as it affects the trigger dates of the process' write-off events. This field becomes protected after the write-off process exists on the database. If you need to change the trigger date on the events you can navigate to the **Events** tab and change the dates manually. Alternatively, you can perform the following "trick":

- Change the Write-off Process Template to something (anything). Doing this causes the **Create Date/Time** to become unprotected.
- Specify the date of the first write-off event in the newly unprotected **Create Date/Time**.
- Change the **Write-off Process Template** back to the desired value. When you change the template, the system removes the events and creates new ones using the **Create Date/Time**.

Enter any **Comments** about the write-off process.

The tree at the bottom of the page shows a variety of information about the **account** and the write-off process including:

- The contracts linked to the write-off process.
- The events linked to the write-off process.
- The amount of debt associated with the account's debt classes.
- A summary of other active collection activities (i.e., collection and write-off processes) associated with the **account**.

Write-off Process - Contracts

The Contract page contains the contracts whose unpaid, finaled debt is being managed by the write-off process. Open this page using **Accounting and Receivable, Write Off Process, Contracts**.

Fastpath: The **Description of Page** section below describes the fields on this page. Refer to How To Perform Common Write-Off Process Functions for more instructions describing how to use this transaction.

Description of Page

The top portion of every page in this page contains the following information:

Write Off Process is a concatenation of summary information about this write-off process. It is composed of the name of the main customer on the account, the write-off status, the reason code for the write-off status, the write-off control name, and the creation date of the process.

ID is the unique system-generated identifier for this write-off process.

Copyright © 2024, Oracle and/or its affiliates.

To modify a contract linked to the write-off process, simply move to a field and change its value. To add a new contract, press + to insert a row, then fill in the information for each field. The following fields display:

Contract identifies the contract being collected. Information about the contract is displayed adjacent.

Write Off Contract Status defines the state of the contract in respect of the write-off process. The following values may exist:

`Active` An `Active` contract is one currently being collected by the write-off process. The contract will remain in this state until the last event linked to the process is complete OR the contract is closed (due to payment in full) OR you want to remove the contract from the write-off process for whatever reason.

`Inactive` A contract becomes `Inactive` when the last event linked to the process is completed OR the contract is closed (due to payment in full) OR you want to remove the contract from the write-off process for whatever reason

The **Write Off Amount** is the amount of debt that initially triggered the write-off process.

The **Contract Balances** area displays how much is owed for the contract. The **Days Old** and **Amount** information shows how old the current balance is. This information will be blank if the current balance is a credit.

Current Balance indicates the amount the customer currently owes for the service. **Payoff Balance** indicates the amount the customer would owe if they wanted to close the contract. This value is only displayed when it differs from the current balance.

Fastpath: Refer to Current Amount versus Payoff Amount for more information.

Write-off Process - Events

The events page contains the activities that will be performed to persuade the customer to pay the outstanding debt. Open this page using **Accounting and Receivable, Write Off Process, Events**.

Fastpath: The **Description of Page** section below describes the fields on this page. Refer to How To Perform Common Write-Off Process Functions for more instructions describing how to use this transaction.

Note: We strongly recommend that you understand the information described in How Are Write-off Events Completed? before using this page.

Fastpath: The number and type of events associated with a write-off process are defined by the process's write-off process template (which is defined on the first page). Refer to Setting Up Write Off Process Templates for more information.

Description of Page

The top portion of every page in this page contains the following information:

Write Off Process is a concatenation of summary information about this write-off process. It is composed of the name of the main customer on the account, the write-off status, the reason code for the write-off status, the write-off class control name, and the creation date of the process.

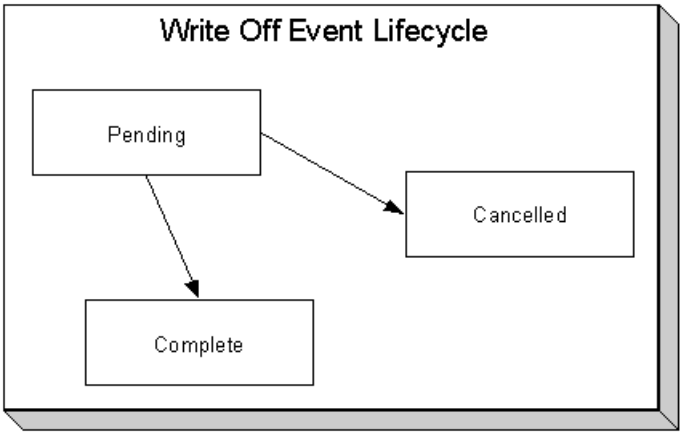
ID is the unique system-generated identifier for this write-off process.

The **Write Off Events** scroll contains individual write-off events that comprise the write-off process. Only one event is displayed at a time. The grid contains the customer contacts that were created by the system when a particular write-off event was triggered (if any).

Event Sequence is the unique identifier of the event.

Write Off Event Status defines the state of the event. The following diagram shows the possible lifecycle of a write-off event:

Copyright © 2024, Oracle and/or its affiliates.



Write-off events are initially created in the `pending` state.

When the system sees a `pending` event with a **Trigger Date** on or before the current date, the system executes the event's activity and `completes` the event.

You can cancel an individual write off event by setting the status to `cancelled`.

The **Write Off Event Type** defines the event's activity (e.g., the related debt is referred to a collection agency, a ToDo entry is generated, a letter is sent).

Fastpath: For more information refer to Setting Up Write Off Process Templates.

The **Trigger Date** defines the date when the system completes (i.e., executes) the event.

Fastpath: For more information about a write-off event's trigger date, see Write-off Event Trigger Date.

If the status of the event is `complete`, the **Completion Date** is displayed with the date on which the system completed the event. If the status of the event is `cancelled`, the **Canceled Date** is displayed with the date on which the event was cancelled.

The **Customer Contact** information contains information when the event template causes a letter to be generated (because letters are created by customer contacts).

How To Perform Common Write-off Process Functions

The topics in this section describe how to perform common write-off process maintenance functions. Refer to The Lifecycle Of A Write-off Process for more information about write-off processes. Refer to The Big Picture Of Write-off Processing for high level information about write-off processing.

How To Create A Write-off Process

99.9% of all write-off processes are created by the Write-Off Monitor and require no human intervention before they are executed (refer to How Are Write-off Processes Created for information about how the account debt monitor creates write-off processes). The other 0.1% are created by users on-line / real time. The following points describe how to create the 0.1%.

- Use **Control Central** to choose the account that requires a new write-off process.
- After the account is populated on Control Central, choose the **Write-off Process +** option on the account context menu to transfer to the write-off process transaction in add mode for the account.
- After the Write-off Process - Main page appears, specify the appropriate **Write Off Control**. **Write Off Control** defines the debt class (i.e., the type of debt) associated with the write-off process. This is important as it controls which contracts can be linked to the write-off process (every contract is associated with a specific debt class). It's

Copyright © 2024, Oracle and/or its affiliates.

important to be aware that if an account's contracts reference multiple write-off debt classes, a write-off process must be created for each write-off debt class that you want to collect.

- Choose a **Write Off Process Template**. **Write Off Process Template** defines the write-off events in the process. You can override these events on the Events page. For more information, refer to Setting Up Write-off Process Templates.
- Navigate to the **Contract** page to define the specific contract's whose debt is covered by this write-off process. You must define at least one contract.
- Save the write-off process.

How To Change Write Off Events

When a write-off process is first created, it has one or more write-off events. The events are the activities that will be performed in the course of the write-off process.

The number and type of events that are created when a write-off process is initiated are defined on the write-off process's write-off process template. The following points describe how to add / change / delete events on a write-off process if the defaulted events are not satisfactory.

- Use **Control Central** to choose the account with the write-off process whose events need to be changed.
- After the account is populated on Control Central, choose the **Write Off Process** option on the account context menu to transfer to the write-off process transaction in update mode for the account.
- To add a new event, transfer to the **Events** tab and press the + button in the **Write Off Events** scroll to add a new event. At this point, the event has not been added to the database; rather, it just exists in memory. Before you add the event to the database, you must specify the following information:
  - Choose an **Event Sequence** so that the new event will be positioned properly in respect of the other events.
  - Choose a **Write Off Event Status** of `Pending`.
  - Choose the desired **Write Off Type Code**.
  - Use **Trigger Date** to define the date on which the event should be activated (i.e., completed).
  - To delete an existing event, transfer to the **Events** tab and use the scroll arrows to toggle to the desired write-off event. When the desired write-off event appears, press the - button to remove the event. At this point, the event has not been removed from the database; rather, it's been removed in memory.
  - To change an existing event, transfer to the **Events** tab and use the scroll arrows to toggle to the desired write-off event. When the desired write-off event appears, make the desired changes.
  - After all desired changes have been made, save the write-off process.

How To Cancel A Write Off Process

The system will automatically "remove" contracts from a write-off process when the close (i.e., when their balance becomes zero). When all contracts are "removed" from a write-off process, the write-off process is cancelled. Refer to How Are Write off Processes and Events Cancelled for more information.

The following points describe how to manually cancel a write-off process.

- Use **Control Central** to choose the account with the write-off process to be cancelled.
- After the account is populated on Control Central, choose the **Write Off Process** option on the account context menu to transfer to the write-off process transaction in update mode for the account. A list of all write-off processes associated with the account appears. If only one write-off process exists, it is automatically selected for you.
- On the **Main** tab, click Cancel.
- When the warning dialog appears, click OK to continue. The write off process and its pending events are canceled.

Writing Off Uncollectable Debt

The Write-off pages described in this section describe the transaction used to manually transfer bad debt to a write-off contract.

Copyright © 2024, Oracle and/or its affiliates.

Write Off - Main

Most write-offs happen behind-the-scenes as part of the automated write-off processing. If you need to perform a partial write-off or cannot wait for the automated process, open **Accounting and Receivable, Write Off**.

Fastpath: Refer to The Big Picture of Write Off Processing for background information about the write-off process.

Description of Page

Account contains the name of the main customer on the account. The **Account ID** is the account whose debt is being written off.

The **Candidate Contracts** scroll contains one row for each of the account's contracts that is eligible for write-off. The system displays the respective age and amount of each contract's debt. The accumulated total of all such debt appears in **Write Off Amount**. The checkbox adjacent to this field is used to confirm that you want to write-off the adjacent debt.

When you click the **Create** button, the system writes off the selected debt by executing the write-off algorithm that's plugged in on the account's customer class. Refer to The Ramifications of Write Offs in the General Ledger for an explanation of how the base-package algorithm works.

How To Manually Write-Off Debt

To manually write-off debt:

- Select the **Account** in question on Write Off - Main.
- Select the debt to be written off for each **Candidate Contract**. You do this by turning on the checkbox adjacent to **Write Off Amount**. You can also vary the amount of the **Write Off Amount**.
- Click the **Create** button.

Write Off - Write Off Contracts

The Write Off Contracts page shows all write-off contracts linked to the account along with their respective current and payoff balance. Use **Accounting and Receivable, Write Off, Write Off Contracts** to open this page.

Description of Page

Account contains the name of the main customer on the account. The **Account ID** is the account associated with the WO contracts displayed below.

The grid contains an entry for every write-off contract linked to the account. The following information is displayed for each write-off contract:

- The contract's **Contract Information** is displayed.
- **Current Balance** contains the WO contract's current balance (i.e., the amount written off).
- **Payoff Amount** contains the WO contract's payoff balance (i.e., the amount written off).

Note: Write off Contract. There is a field on contract type called Special Role Flag. Only those contract types with a role of `Write Off` are displayed.

Collection Referral

The Collection Referrals page contains information about an account's debt that has been referred to a collection agency.

Fastpath: Refer to How Do Collection Agency Referrals Work? for more information about how the system creates and maintains collection agency referrals as part of the automated write off processing. In theory, you need only access this page if you need to override the automated processing.

Copyright © 2024, Oracle and/or its affiliates.

Open **Delinquency Management, Collection Agency Referral** to maintain this information.

Description of Page

The **Agency Referrals** scroll contains one entry for every collection agency referral associated with the account. The following information is defined for each referral.

Use **Collection Agency** to define the agency to which the referral is being sent.

Start Date is the date on which the referral was initially created.

Referral Status defines if the referral is `Active` or `Closed`.

Use **Comments** to describe anything unusual about the referral.

The grid contains the history of interactions with the collection agency. Each time an account's debt is referred to a collection agency, the system creates a referral history record.

You can communicate changes about the referral by inserting a new row in the collection. For example,

- If you need to change the referral amount, insert a row and indicate the **Creation Date** and a **Referral History Reason** of `Change Referral`.
- If you need to cancel the referral, insert a row and indicate a **Referral History Reason** of `Referral Cancellation`.
- If the customer pays, insert a row and indicate a **Referral History Reason** of `Referral Paid`.
- If you need to add a new referral, insert a row and indicate a **Referral History Reason** of `Initial Referral`.

Collection agency referral records are interfaced to the respective collection agency using the batch process defined on the collection agency control record (refer to Setting Up Collection Agencies). The **Batch Control** process and the respective **Batch Number** in which the records were interfaced to the collection agency are displayed adjacent.

The **System Generated** switch is on for those collection agency referrals created as part of the automated write-off processing. Refer to How Do Collection Agency Referrals Work? for more information.

Payment Arrangements

Payment arrangements is used to payoff debt. It is an agreement with a customer to pay off their debt as part of their future bills (sometimes referred to as "current bill plus").

Setting Up Payment Arrangements

A payment arrangement is an agreement with a customer to payoff severely overdue debt in installments. Bills sent to customers with payment arrangements contain charges for both their current services and their payment arrangement installment amount. Refer to The Big Picture Of Payment Arrangements for information about the financial ramifications of payment arrangements and for a description of how the system monitors payment arrangements.

The topics in this section describe how to set up a payment arrangement and how to break a payment arrangement (if a customer doesn't fulfill their obligations).

Pay Arrangement - Main

This page allows you to set up a payment arrangement. When you create a payment arrangement, the system performs the following functions:

- It creates a payment arrangement contract (PA contract). It sets the installment amount on this contract equal to the installment amount specified on this page.
- It transfers delinquent debt from each delinquent contract(s) to the new PA contract.
- As it transfers the debt to the PA contract, it creates synchronizing adjustments to reduce the PA contract's current balance by the amount transferred. Why? Because when delinquent debt is transferred to the PA contract, its current balance increases. Because current balance contains the amount the customer currently owes, this balance must be

Copyright © 2024, Oracle and/or its affiliates.

reduced because the customer is going to payoff the debt in installments (and therefore they don't currently owe anything on the PA contract).

You could do the above functions by adding a new contract (using the Contract page) and creating transfer adjustments (using the adjustment page). However, this is tedious. Rather, open **Delinquency Management, Pay Arrangement** to set up or add additional debt to a payment arrangement.

Description of Page

Contract Info and **Contract ID** are displayed on every page. These values only appear after the payment arrangement contract (PA contract) exists on the database. The **ID** is a system assigned random number that stays with the payment arrangement contract for life. The **Contract Info** is a concatenation of important details about the payment arrangement contract and its account.

The **Current Balance** that appears beneath **Contract Info** contains the *PA Contract's* current balance. When you initially set up a PA contract this balance will be zero. It is only non-zero if the customer has not paid a billed installment.

Payoff Balance beneath **Contract Info** contains the *PA Contract's* payoff balance. The payoff balance is the total amount to be paid off over the PA contract's life. It is only non-zero after delinquent funds have been transferred to the PA contract. This balance is only displayed when it differs from the **Current Balance**.

Fastpath: Refer to Current Amount versus Payoff Amount for more information.

Account contains the name of the main customer on the account. **Account ID** is the account associated with the PA contract.

The occurrences of **Current** and **Payoff Balance** below **Account** are the *account's* respective balances. These values are displayed to help you confirm exactly how much the customer currently owes versus how much they will have to payoff over time (remember, after you transfer funds to a payment arrangement, the amount transferred is reduced from the account's current balance).

The **Candidate Contracts** scroll contains one row for each of the account's contracts with aged debt. The system displays the respective age and amount of each contract's debt. All debt that is checked will be transferred to the PA contract when you click **Create**.

CAUTION: The system automatically indicates that all debt that isn't "new" (i.e., debt that the customer has seen on their bills) should be transferred to the payment arrangement. It does this by checking each parcel of aged debt.

Total is the sum of ALL current balances (not just the checked amounts) from all candidate contracts.

The next two fields are used to populate the arrangement amount (i.e., the amount the customer will be billed on each bill).

- The **Installments** field exists to calculate the **Arrange Amount** (the next field) when a time period has been stipulated. Simply enter the number of installments and tab out. When you tab out of the field, the system sets the **Arrange Amount** equal to **New Payoff Balance**/ **Installments**.

Note: Arrange Amount may not add up. You will note that the **Arrange Amount** is rounded up so that if you multiply the number of installments by the calculated installment amount, the result will be greater than the PA contract's payoff amount. Don't worry - only the actual dollar amount of the PA contract's total debt will be billed by the system. The last installment will be slightly lower than the other installments.

- Use the **Arrange Amount** field to define the payment arrangement's installment amount if a specific dollar amount has been stipulated. For example, if the customer agrees to pay off their delinquent debt by paying an additional $50 on each bill, you'd enter "50" in this field. Note - If you enter a dollar amount in this field and tab out, the system will calculate the number of installments and display the number in the **Installments** field.

New Payoff Balance contains the PA contract's payoff balance. This is the total amount of debt that will be paid off over the lifetime of the PA contract. This value equals the current payoff balance on the PA contract plus all selected debt from the **Candidate Contracts** scroll.

Define the **Division** and **Contract Type** of the payment arrangement contract.

Copyright © 2024, Oracle and/or its affiliates.

> **Note: Payment arrangement contract types.** There is a field on contract type called Special Role Flag. Only those contract types with a role of `Payment Arrangement` may be selected.

Clicking **Create** (or **Change**)causes the system to perform the following:

- It adds / updates a payment arrangement contract (PA contract). It sets the installment amount on this contract equal to the installment amount specified on the page.
- It transfers delinquent debt from the delinquent contract(s) to the PA contract. The adjustment type used to transfer these funds is defined on the PA contract's contract type.
- It reduces the current balance on the PA contract by the amount of debt transferred to the PA contract. Why? Because when delinquent debt is transferred to the PA contract, its current balance increases. Because current balance contains the amount the customer currently owes, this balance must be reduced because the customer is going to pay off the debt in installments (and therefore they don't currently owe anything). The adjustment type used to transfer these funds is defined on the PA contract's contract type.

You can transfer to the **History** page to view the transfer adjustments that were created by the system when it transferred the selected debt to the PA contract.

Clicking **Break** causes the PA contract to become broken. Please refer to the note below for what transpires when a PA is broken.

> **Note: Break logic is in a plug-in.** Please be aware that the logic that is executed when a payment arrangement is broken exists in a plug-in algorithm (plugged-in on the PA contract's contract type). The base package algorithm expires the PA contract and cancels all related frozen adjustments, returning the debt to the original contracts. Because this logic is in a plug-in, you can develop alternative logic and plug it in if the base package logic is not satisfactory.

How To Set Up A Payment Arrangement

To set up a new payment arrangement:

- Select the **Account** in question.
- Select the debt to be transferred to the payment arrangement (by turning on/off the check boxes adjacent to the **Candidate Contract's** amounts).
- Enter the number of **Installments** and the total **Arrange Amount**.
- Define the **Division** and **Contract Type** of the new payment arrangement contract.
- Click the **Create** button.

How To Add Additional Debt To A Payment Arrangement

To transfer additional debt to an existing payment arrangement:

- Select the **Contract ID** of the existing payment arrangement.
- Select the debt to be transferred to the payment arrangement (by turning on/off the checkboxes adjacent to the **Candidate Contract's** amounts).
- Enter the number of **Installments** and the total **Arrange Amount**.
- Click the **Create** button.

> **Note: After transfer.** After transferring the funds to the payment arrangement contract, confirm the account's payoff and current balances make sense (the payoff balance is the total amount of debt that the account will eventually have to payoff, the current balance is how much they currently owe, i.e., the amount that credit and collections monitors). The account's payoff and current balances are displayed in the page's second section. If you do not understand the difference between payoff balance and current balance, refer to Current Amount versus Payoff Amount.

How To Break A Payment Arrangement

You would break a payment arrangement when a customer doesn't make the agreed payments. Typically, your credit and collections processes will break a payment arrangement behind-the-scenes. Refer to The Big Picture Of Payment Arrangements for information about how the system monitors payment arrangements.

Copyright © 2024, Oracle and/or its affiliates.

If you want to manually break a payment arrangement simply select the **Contract ID** of the existing payment arrangement and click the **Break** button.

How To Cancel A Payment Arrangement

You would cancel a payment arrangement if you made a mistake and never intended to set up the payment arrangement in the first place. To cancel a payment arrangement

- Select the **Contract ID** of the existing payment arrangement and navigate to the **History** tab.
- Click the **Cancel** button to cancel the transfer adjustments that were used to transfer debt to the payment arrangement from the customer's regular contracts.
- Next, navigate to the payment arrangement contract's (PA contract) financial history page (using the contract context menu). You will see that there are a few non-canceled adjustments (these were used to change the PA contract's current balance after debt was transferred to the PA contract). Cancel these adjustments by drilling down on these adjustments and then click the **Cancel** button on the adjustment maintenance transaction.
- Finally, display the PA contract on the contract maintenance transaction and cancel the payment arrangement.

Pay Arrangement - History

Open **Delinquency Management, Pay Arrangement, History** to view the transfer adjustments that were created by the system when it transferred the selected debt to the PA contract.

Description of Page

Contract (contract) **Info** and **ID** are displayed on every page in this page. These values only appear after the payment arrangement contract (PA contract) exists on the database. The ID is a system-assigned random number that stays with the contract for life. The Contract Info is a concatenation of important details about the contract and its account.

Current Balance beneath **Contract Info** contains the *PA Contract's* current balance. When you initially set up a PA contract this balance will be zero. It is only non-zero if the customer has not paid a billed installment.

Payoff Balance beneath **Contract Info** contains the *PA Contract's* payoff balance. The payoff balance is the total amount to be paid off over the PA contract's life. It is only non-zero after delinquent funds have been transferred to the PA contract. This balance is only displayed when it differs from the Current Balance.

Fastpath: Refer to Current Amount versus Payoff Amount for more information.

Account contains the name of the main customer on the account. **Account ID** is the account associated with the PA contract.

The occurrences of **Current** and **Payoff Balance** below **Account** are the *account's* respective balances. These values are displayed to help you confirm exactly how much the customer currently owes versus how much they will have to payoff over time (remember, after you transfer funds to a payment arrangement, the amount transferred is reduced from the account's current balance).

The **Adjustment Details** area contains one row for each transfer adjustment used to transfer delinquent debt to the PA contract. Click the Go To button to transfer to the adjustment page to view the details of an adjustment (or to cancel the adjustment if you want to reinstate the debt on the originating contract).

Total Frozen Adj is the total amount of debt transferred to the payment arrangement that has not been canceled.

Clicking **Cancel** causes the system to cancel all related frozen adjustments, returning the debt to the original contracts. You must specify the cancel reason by selecting it from the drop down list when Payment Arrangement Cancellation window is displayed.

Total Canceled Adj is the total amount of debt transferred to the payment arrangement that has been canceled.

Copyright © 2024, Oracle and/or its affiliates.

Promise To Pay

The **Promise To Pay** screen allows you to search promise to pay. You can also add and remove schedule payments. This screen consists of the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Promise To Pay ID	Used to specify promise to pay ID.	Yes
Account ID	Used to specify account ID for the promise to pay.	Yes
Status	Displays the status of promise to pay.	Not applicable
Created By	Displays the user ID who has created the promise to pay.	Not applicable
Last Updated by	Displays the user ID who has last updated the promise to pay.	Not applicable
Type	Displays the promise to pay type.	Yes
Debt Class	Displays the debt class of the account associated with the promise to pay.	Not applicable
Current Balance	Displays the current balance of the account associated with the promise to pay.	Not applicable
Third Party Payor	Select the **Third Party Payor** check box if the payments for the promise to pay will be made by a third party other than the person.	Yes
Payor Account ID	Indicates account ID of the payor.	Not applicable
Start Date	Displays the start date of the promise to pay.	Yes
Pay Method	Indicates the payment method for the promise to pay.	Yes

In addition this screen the **Scheduled Payments** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Scheduled Date	Used to specify the date when the payment is expected as per promise to pay.	Yes
Scheduled Amount	Used to specify amount to be paid on the **Scheduled Date**.	Yes
Clearing ID	Used to display the clearing ID. **Note:** This field is displayed when the status of the Promise to Pay is **Active**.	Not applicable
Auto Pay ID	Used to specify Auto Pay ID. **Note:** The auto pay ID can be specified only if auto-pay exists on the account.	Yes
Total Amount	Displays the total amount payable as per the promise to pay.	Not applicable

Copyright © 2024, Oracle and/or its affiliates.

Note: You can add multiple schedule records with same **Scheduled Date** but different **Auto Pay ID**.

This screen contains the following buttons:

Button Name	Button Description
Cancel	Used to cancel the promise to pay.
	Note: The Promise to pay can be cancelled when it is in **Pending** status.

Search

The **Search** zone allows you to search for promise to pay. This zone contains the following two sections:

- **Search Criteria** - The **Search Criteria** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Person Name	Used to search for promise to pay which belong to a particular person.	No
Promise To Pay ID	Used to search promise to pay with promise to pay ID.	No
	Note: The promise to pay ID is generated automatically after promise to pay is created.	
Account ID	Used to search for a particular account.	No

Note: You must specify at least one search criterion while searching for a promise to pay.

- **Search Results** - On clicking the **Search** button, the search results appear based on the specified search criteria. The **Search Results** section contains the following columns:

Column Name	Column Description
Person Name	Displays the name of the person associated with the promise to pay.
Promise To Pay ID	Displays the promise to pay ID.
Account ID	Displays the account ID associated with promise to pay.
Start Date	Displays the start date of the promise to pay.
Promise To Pay Status	Displays the status of promise to pay.
Debt Class	Displays the debt class of the account associated with the promise to pay.

Related Topics

For more information on...	See...
Promise To Pay screen	Promise To Pay on page 543
How to search for a promise to pay	Searching a Promise To Pay on page 545
How to define a promise to pay	Adding a Promise To Pay on page 545
How to edit a promise to pay	Editing Promise To Pay on page 546

Copyright © 2024, Oracle and/or its affiliates.

Searching a Promise To Pay

Procedure

To search a promise to pay:

1. Click the **Menu** link in the **Actions/Navigation** area.
   A list appears.
2. Select the **Main Menu** option from the list.
3. From the **Main Menu**, select **Delinquency Management** and then click **Promise To Pay**.
   The **Promise To Pay Search** screen appears.
4. Enter the search criteria in the **Search** zone.
5. Click the **Search**.
   The screen displays the filtered results.

Related Topics

For more information on...	See...
Promise To Pay screen	Promise To Pay on page 543
How to add a promise to pay	Adding a Promise To Pay on page 545
How to edit a promise to pay	Editing Promise To Pay on page 546
How to cancel a promise to pay	Cancelling Promise To Pay on page 548

Adding a Promise To Pay

Procedure

To add promise to pay:

1. Click the **Menu** link in the **Application** toolbar.
   A list appears.
2. From the **Main** menu, select **Delinquency Management** and then click **Promise To Pay**.
   A sub-menu appears.
3. Click the **Add** option from the **Promise To Pay** sub-menu.
   The **Promise To Pay** screen appears. It contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Promise To Pay ID	Used to specify promise to pay ID.	Yes
Account ID	Used to specify account ID for the promise to pay.	Yes
Status	Displays the status of promise to pay.	Not applicable
Created By	Displays the user ID who has created the promise to pay.	Not applicable
Last Updated by	Displays the user ID who has last updated the promise to pay.	Not applicable
Type	Displays the promise to pay type.	Yes

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
Debt Class	Displays the debt class of the account associated with the promise to pay.	Not applicable
Current Balance	Displays the current balance of the account associated with the promise to pay.	Not applicable
Third Party Payor	Select the **Third Party Payor**check box if the payments for the promise to pay will be made by a third party other than the customer.	Yes
Payor Account ID	Indicates account ID of the payor.	Not applicable
Start Date	Displays the start date of the promise to pay.	Yes
Pay Method	Indicates the payment method for the promise to pay.	Yes

In addition this screen the **Scheduled Payments** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Scheduled Date	Used to specify the date when the payment is expected as per promise to pay.	Yes
Scheduled Amount	Used to specify amount to be paid on the **Scheduled Date**.	Yes
Auto Pay ID	Used to specify Auto Pay ID.	Yes
Total Amount	Displays the total amount payable as per the promise to pay.	Not applicable

This screen contains the following buttons:

Button Name	Button Description
Cancel	Used to cancel the promise to pay.

4. Enter the required details.
5. Click **Save**.
   The promise to pay is added.

Related Topics

For more information on...	See...
Promise To Pay screen	Promise To Pay on page 543

Editing Promise To Pay

Procedure

To edit promise to pay:

1. Click the **Menu** link in the **Application** toolbar.
   A list appears.
2. From the **Main Menu**, select **Delinquency Management** and then click **Promise To Pay**.
   The **Promise To Pay** screen appears. It contains the following fields:

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
Promise To Pay ID	Used to specify promise to pay ID.	Yes
Account ID	Used to specify account ID for the promise to pay.	Yes
Status	Displays the status of promise to pay.	Not applicable
Created By	Displays the user ID who has created the promise to pay.	Not applicable
Last Updated by	Displays the user ID who has last updated the promise to pay.	Not applicable
Type	Displays the promise to pay type.	Yes
Debt Class	Displays the debt class of the account associated with the promise to pay.	Not applicable
Current Balance	Displays the current balance of the account associated with the promise to pay.	Not applicable
Third Party Payor	Select the **Third Party Payor** check box if the payments for the promise to pay will be made by a third party other than the customer.	Yes
Payor Account ID	Indicates account ID of the payor.	Not applicable
Start Date	Displays the start date of the promise to pay.	Yes
Pay Method	Indicates the payment method for the promise to pay.	Yes

In addition this screen the **Scheduled Payments** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Scheduled Date	Used to specify the date when the payment is expected as per promise to pay.	Yes
Scheduled Amount	Used to specify amount to be paid on the **Scheduled Date**.	Yes
Auto Pay ID	Used to specify Auto Pay ID.	Yes
Total Amount	Displays the total amount payable as per the promise to pay.	Not applicable

This screen contains the following buttons:

Button Name	Button Description
Cancel	Used to cancel the promise to pay.
	Note: The promise to pay can be cancelled when it is in **Pending** status.

3. Modify the required details.
4. Click **Save**.
   The changes made to promise to pay are saved.

Related Topics

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
Promise To Pay screen	Promise To Pay on page 543
How to define a promise to pay	Adding a Promise To Pay on page 545
How to cancel a promise to pay	Cancelling Promise To Pay on page 548

Cancelling Promise To Pay

Procedure

To cancel a promise to pay:

1. Click the **Menu** link in the **Actions/Navigation** area.
   A list appears.
2. Select the **Main Menu** option from the list.
3. From the **Main Menu**, select **Delinquency Management** and then click **Promise To Pay**.
   The **Promise To Pay** screen appears.
4. Click **Cancel**.
   The **Cancel Reason** window appears. It contains following fields:

Field Name	Field Description	Mandatory (Yes or No)
Cancel Reason	Used to specify the reason why you want to cancel the promise to pay.	Yes

5. Select the rejection reason from the **Cancel Reason** list.
6. Click **OK**.

Related Topics

For more information on...	See...
Promise To Pay screen	Promise To Pay on page 543

Chapter 6

Financial Transactions

Topics:

- The Big Picture Of Financial Transactions
- Financial Transaction
- Account Financial History
- Account Bill/Payment History
- Contract Financial History
- Contract Cash Accounting Balance
- Balance Control
- Match Event
- Billed Revenue
- FT GL Drill Down Details

In this section, we describe the financial transactions generated as a result of your bills, payments and adjustments.

The Big Picture Of Financial Transactions

The topics in this section provide background information about a variety of financial transaction issues.

Fastpath: We strongly recommend familiarizing yourself with the topics described in The Financial Big Picture to fully appreciate the system's financial architecture.

Bill Segment Financial Transactions

A bill segment has a related financial transaction. The financial transaction contains the financial effects of the bill segment on the contract's current and payoff balances and on the general ledger.

If a bill segment is cancelled, another financial transaction is created to reverse the original financial transaction. The cancellation financial transaction appears on the next bill produced for the account as a bill correction.

Fastpath: For more information about bill segment financial transactions, refer to Bill Details.

Payment Segment Financial Transactions

A payment segment has a related financial transaction. The financial transaction contains the financial effects of the payment segment on the contract's current and payoff balances and on the general ledger.

If a payment segment is cancelled, another financial transaction is created to reverse the original financial transaction. The cancellation financial transaction appears on the next bill produced for the account as a negative payment.

Fastpath: For more information about payment segment financial transactions, refer to Payment Details.

Adjustment Financial Transactions

An adjustment has a related financial transaction. The financial transaction contains the financial effects of the adjustment on the contract's debt and on the general ledger.

If the adjustment is eventually cancelled, another financial transaction will be linked to the adjustment to reverse its financial effect. The cancellation financial transaction appears on the next bill produced for the account as an adjustment.

Fastpath: For more information about adjustment financial transactions, refer to Adjustment Details.

Financial Transactions And Aged Debt

When a financial transaction's related bill segment / payment segment / adjustment is frozen, the financial transaction (FT) is also frozen. When an FT is frozen, its contract's debt is impacted. It is important to stress the following in respect of this impact:

- The FT's GL details will be interfaced to the GL when the GL interface next executes.
- If the FT decreases the amount of debt, the customer's aged debt is affected immediately regardless of whether the FT appears on a bill.
- If the FT increases the amount of debt, the amount the customer owes from an *aged debt* perspective may or may not be affected by the FT. There is a switch on an FT called New Charge that controls the arrears behavior. If this switch is on, the customer's aged debt will not reflect the FT amount until the FT is swept onto a bill. The moment the FT is swept onto the customer's bill, the debt starts aging. If this switch is off, the date on which the FT starts aging must be defined in the Arrears Date field.

Copyright © 2024, Oracle and/or its affiliates.

- The amount a customer owes *in total* is immediately affected by the FT regardless of whether the FT appears on a bill. This means that the amount of aged debt may not be in sync with the total amount owed. This seems odd, but is useful from a credit and collections perspective. You see, you probably don't want to start aging a FT until the customer has actually seen it.

Fastpath: For more information, refer to Financial Transactions Created Between Bills.

Current Balance versus Payoff Balance

Fastpath: For information about current and payoff balance in general, refer to Current Amount versus Payoff Amount.

The topics in this section describe when payoff amount differs from current amount for the various types of financial transactions.

Adjustments - Current Balance versus Payoff Balance

Fastpath: For information about current and payoff balance for adjustments, refer to Adjustments - Current Balance versus Payoff Balance.

Billing - Current Balance versus Payoff Balance

Fastpath: For information about current and payoff balance for bill segments, refer to Billing - Current Balance versus Payoff Balance.

Payment - Current Balance versus Payoff Balance

Fastpath: For information about current and payoff balance for payment segments, refer to Payment - Current Balance versus Payoff Balance.

The Source Of GL Accounts On Financial Transactions

Fastpath: For information about the source of the distribution codes used to generate the GL accounts on the financial transactions, refer to The Source Of GL Accounts On Financial Transactions.

Obscure Things That Can Happen

The topics in this section provide information about obscure things that can happen when a financial transaction (FT) is frozen.

A Stopped Contract May Be Closed

If an FT causes a `Stopped` contract's current and payoff balances to become zero, the system closes the contract (i.e., the contract's status becomes `Closed`).

A Closed Contract May Be Reactivated

After a contract is closed (i.e., after it's stopped and paid-in-full), it's possible for some types of financial transactions to be linked to the contract. These financial transactions could cause the current and/or payoff balances to become non-zero. If this happens, the system reactivates the contract (i.e., the contract's status becomes `reactivated`).

When a contract becomes reactivated, it becomes eligible for review by the write-off process monitor (when it next runs). The write-off process monitor will, in all likelihood, start a write-off process for the reactivated contract.

Copyright © 2024, Oracle and/or its affiliates.

Financial events that can be linked to a closed contract are:

- The cancellation of a bill segment
- The freezing of a payment segment
- The cancellation of a payment segment
- The freezing of an adjustment
- The cancellation of an adjustment

A Write-Off Process May Be Deactivated If The Contract Is Closed

When a contract becomes `Closed`, it will be removed from any ongoing write-off processes. This removal could cause the write-off process to become `Inactive` (if this is the last contract linked to the write-off process). The inactivation of the write-off process, could, in turn, cause collection agency referrals associated with the write-off process to be cancelled.

A Reactivated Contract May Be Closed

After a contract has been reactivated (refer to the previous section for how this happens), a financial transaction (or transactions) may be linked to it that will cause the current and payoff balance to return to zero. If this happens, the system closes the contract (i.e., the contract's status becomes `Closed`).

All types of financial transactions can be posted to a `Reactivated` contract.

One Or More Algorithms May Be Executed

If the FT is linked to a contract whose contract type has Freeze Algorithms, these algorithms will be executed when the FT is frozen. Because you can code whatever you please in an algorithm, anything could happen. Examples of such algorithm include:

- An algorithm that routes product's consumption to third party service providers when a bill is frozen (refer to STG SND CONS for more information). This type of algorithm would be plugged in on contract types whose product's consumption can be interfaced to other service providers.
- A bill message can be added to the account when the system detects that its deposit has been paid in full (refer to DEP PIF MSG for more information). This type of algorithm would be plugged in on deposit contract types.

In addition, FT freeze algorithms can also be defined on an account's customer class. Because you can code whatever you please in an algorithm, anything could happen. An example of such an algorithm is one that cancels match events when an FT is cancelled (refer to CNCL FT MEVT for more information). This type of algorithm would be plugged in on customer classes associated with open item customers.

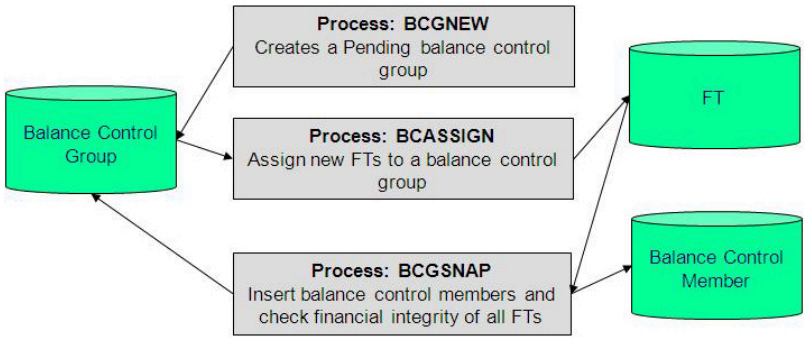
The Big Picture of Balance Control

The balance control processes are used to check the financial integrity of your system. The contents of this section describe how these processes work.

The Balance Control Background Processes

The following diagram illustrates the balance control background processes:

Copyright © 2024, Oracle and/or its affiliates.



BCGNEW - Create A New Balance Control Group

This process creates a `Pending` balance control group if one doesn't already exist. You may wonder why an entire process is dedicated to such a trivial task. The reason is because the next process, BCASSIGN, is a multi-threaded (i.e., parallel) process and we only need one `Pending` balance control group regardless of the number of threads used to assign balance control ID's to financial transactions.

BCASSIGN - Assign New Financial Transactions A Balance Control Group

This multi-threaded (i.e., parallel) process assigns the `Pending` balance control group to new FTs whose freeze date/time is before the create date/time of the balance control record.

BCGSNAP - Insert BC Members And Check Financial Integrity Of All FTs

This process performs the following two functions:

- It summarizes new financial transactions under the current Pending balance control group as follows:
- - It creates a balance control member for every combination of **Division**, **Contract Type** and **FT Type** referenced on the financial transactions belonging to the balance control group.
  - It updates each balance control member with the following information:
- The number of financial transactions (FTs)
- The sum of the total amounts on the FTs in this balance control group.
- The sum of the current amounts on the FTs in this balance control group.
- The sum of the total amounts from all FTs (in this and all other balance control groups).
- The sum of the current amounts from all FTs (in this and all other balance control groups).

  - It sets the status of the balance control group to `Complete`.
  - It checks the integrity of the financial transactions in each historical Balance Control Group. It does this by summarizing EVERY financial transaction throughout time and determining if the sums are in sync with the values maintained on the balance control members. If integrity problems are detected, a detailed error message is displayed on the run control associated with the process.

If you opt to run the balance control processes on a nightly basis, you will find that the verification processing will take longer every night (because there are more financial transactions over time). In order to deal with this issue, the BCGSNAP process has a parameter that allows you to control which of the above functions is implemented (it's call VERIFY-ONLY-SW). In order to speed nightly processing, run this process with the switch set to "G" (this causes new financial transactions to be summarized under a new balance control). Then, once a week (or month), run this process with the switch set to "Y" (this checks the financial integrity of all financial transactions in the system).

If you run the balance control processes less frequently, you can set the VERIFY-ONLY-SW to "N" (this causes both of the above functions to execute).

Consider A Backup At This Point

We recommend you backup your database AFTER you run the above processes. Why? So that if a financial integrity problem is spotted in the future, you can compare the current database against the backup to see what changed.

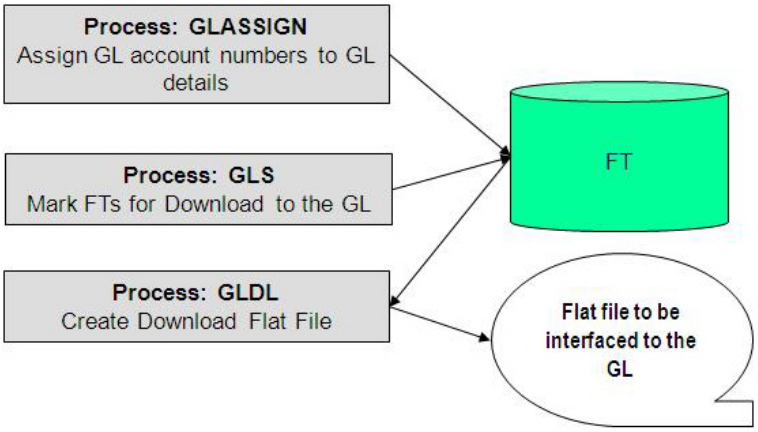
Copyright © 2024, Oracle and/or its affiliates.

Balance Control Information Is Available Online

Your internal auditors may be interested in the total number and amount of financial transactions that have been posted since a given point in time. You can use the Balance Control page to see this information.

The GL Interface

The following diagram illustrates the GL Interface.



GLASSIGN - Assign GL Account Numbers To GL Details

The `GLASSIGN` process assigns GL account numbers to the GL details associated with financial transactions. GL account numbers are assigned as follows:

- Every GL detail references a distribution code.
- Every distribution code references a GL assignment algorithm. The base package algorithm simply uses the default GL account associated with the distribution code. However, you can construct your own algorithm(s) to assemble your GL account numbers in your desired fashion. Refer to Setting Up Distribution Codes for more information about the GL format algorithm.
- The `GLASSIGN` process simply calls each GL's details distribution code's GL assignment algorithm and updates the GL detail with the result (i.e., the GL account number). This GL account number is then used by the `GLDL` process when it creates the consolidated journal entry that's interfaced to the GL.

Note: If incorrect GL account numbers get assigned to GL details... If you do not plug in the correct algorithm or your algorithm is wrong, you can correct the GL account numbers that are assigned to the GL details. How? Write a simple program that resets the GL account numbers on the erroneous GL details. Then run `GLASSIGN`. `GLASSIGN` will re-execute the distribution code GL assignment algorithm and refresh the account number accordingly.

GLS - Prepare FTs for Download

The `GLS` process creates FT download staging records for all FTs that are ready to be posted to the GL (the FT download staging records are stored on the FT/Process table). Each FT download staging record is marked with a batch process ID and run number.

- The batch process ID is the process responsible for creating the flat file that contains the consolidated journal entry that is interfaced to your general ledger. The batch process ID is defined on Installation Options - Financial. The system comes with a single GL download program (GLDL - Create General Ledger Download Flat File). The GL account numbers are provided by the GL account algorithms that are specified on the distribution codes. Refer to Setting Up Distribution Codes for more information about the GL account algorithm.
- The run number is the batch process ID's current run number.

This process also changes the status of the FT to `distributed`. You may not change the FT's GL details after this time.

Copyright © 2024, Oracle and/or its affiliates.

GLDL - Create General Ledger Download Flat File

The `GLDL` process creates the flat file that contains the consolidated journal entry that is interfaced to your general ledger. One header and multiple detail records are created as described below. At the conclusion of the batch process, a validation is performed to compare debits against credits. If they are not equal, the process terminates with an error.

Header Record Layout

The following table lists the fields that are extracted in the header of a flat file which is generated using the **Download Extract (GLDL)** batch:

Field Name	Format	Source/Value/Description
REC_TYPE	A1	1 (1 is used for header records)
BATCH_CD	A8	The code for the batch process that created the file. (This value will always be 'GLDL'.)
BATCH_NBR	N10	The batch process will be run many times. This field indicates which of those runs produced a particular output file. This may be thought of as an instance identifier for the batch process.
BATCH_RERUN_NBR	N10	Any given instance of the GLDL batch process may be re-run at any time. If this instance has been re-run, this field will be populated with a number indicating how often.
EXTRACT_DTTM	A26	The date and time of the run that created a particular output file.
DETAIL_REC_CNT	N12	The number of details records
DETAIL_REC_TOTAL_DR	N13.2	This is the sum of FINANCIAL_AMOUNTs from all detail records that were greater than zero (debits).
DETAIL_REC_TOTAL_CR	N13.2	This is the sum of FINANCIAL_AMOUNTs from all detail records that were less than zero (credits).

Detail Record Layout for GLDL

Note: A record is created for each unique occurrence of REC_TYPE, GL_DIVISION, CURRENCY_CD, GL_ACCOUNT and ACCT_PERIOD. If a given GL account has debit and credit FINANCIAL_AMOUNTs, two records will be created - one shows the debit amount, the other shows the credit amount.

The following table lists the fields of the FT GL records which are extracted while executing the **GL Download Extract (GLDL)** batch:

Field Name	Format	Source/Value/Description
REC_TYPE	A1	2 (2 is used for detail records)
GL_DIVISION	A5	This is the GL Division from the CI_FT record. It determines a) which Accounting Calendar will be used to calculate the Accounting Period (below) and b) the currency of the FT.
CURRENCY_CD	A3	The currency in which the Amount is denominated.
GL_ACCOUNT	A254	Contains the GL account number as supplied by the distribution code's GL account algorithm.
ACCT_PERIOD	A6	An accounting period in the format YYYYPP where YYYY is the fiscal year and PP is the accounting period. This is

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
		derived from the GL detail's FT's accounting date and GL division.
FINANCIAL_AMOUNT	N13.2	This is the debit or credit amount. Debits are represented by positive numbers; credits are represented by negative numbers. If a given GL account has debit and credit entries, two records will be created - one shows the debit amount, the other shows the credit amount.
STAT_CODE	A8	This is the GL distribution code's statistic code (if any)
STAT_AMOUNT	N13.2	This is the statistical amount from the GL detail lines

Note: The GLASSIGN process updates an FT's GL details with the appropriate GL account number.

GL Posting for Zero Valued Transaction

Until now, the system did not create FT GL entries for a zero amount financial transaction (i.e. bill segment, payment, adjustment, etc.) Now, the system enables you to create dual FT GL entries for a zero amount financial transaction. A new option type named **Allow non-significant zero amount FT GL entry** is added in the **C1_FINTRANOP** feature configuration. To enable dual FT GL creation for a zero amount financial transaction, you need to set the **Allow non-significant zero amount FT GL entry** option type of the **C1_FINTRANOP** feature configuration to **Y**. The system considers the FT GL amount as significant when the statistical and total amount of the financial transaction is non-zero.

Different GL Accounting Currency

Until now, the system allowed you to create additional FT GL entries for a financial transaction in the **CI_FT_GL_EXT** table when the financial transaction currency was different from the division's base currency. The system enabled you to create additional FT GL entries for a financial transaction in either of the following two ways:

- While freezing a financial transaction from the user interface
- While assigning a GL account to a financial transaction through the **Assign GL Account to Financial Transaction (C1-GLASN)** or **Assign GL Account to Financial Transaction (GLASSGN2)** batch

There might be scenarios wherein an organization might want to create additional FT GL entries in the GL's reporting currency which is different from the division's base currency. Oracle Revenue Management and Billing enables you to define local currency for a division. A new field named **Local Currency** is added in the **Main** tab of the **Division** screen. You must specify the local currency only when it is different from division's base currency. It is not a mandatory field. You can also add the division's local currency in the **Invoice Currencies** tab and define its rounding method for accurate calculations. A new field named **Enable Local Currency** is available while defining a distribution code. It provides the ability to enable the FT GL entry creation in the local currency for the required GL account.

Now, while creating additional FT GL entries for a financial transaction, the system checks whether the financial transaction currency is different from the division's base currency or division's local currency (if specified). If the financial transaction currency is different from the division's base currency and if the local currency is not defined for the division or if the financial transaction currency is same as the division's local currency, the system creates additional FT GL entries for the financial transaction in the division's base currency. If the financial transaction currency is same as the division's base currency but if the financial transaction currency is different from the division's local currency, the system checks whether the **Enable Local Currency** option is selected in the respective distribution code.

If the **Enable Local Currency** option is selected in the respective distribution code, the system creates additional FT GL entries for the financial transaction in the division's local currency. However, if the **Enable Local Currency** option is not selected in the respective distribution code, the system does not create additional FT GL entries for the financial transaction in the **CI_FT_GL_EXT** table.

Copyright © 2024, Oracle and/or its affiliates.

If the financial transaction currency is different from the division's base currency and division's local currency, the system checks whether the **Enable Local Currency** option is selected in the respective distribution code. If the **Enable Local Currency** option is selected in the respective distribution code, the system creates additional FT GL entries for the financial transaction in the division's base and local currencies. However, if the **Enable Local Currency** option is not selected in the respective distribution code, the system creates additional FT GL entries for the financial transaction in the division's base currency.

However, if the financial transaction currency is same as the division's base currency and if the local currency is not defined for the division or if the financial transaction currency is same as the division's local currency, the system does not create additional FT GL entries for the financial transaction in the **CI_FT_GL_EXT** table.

Earlier, you were able to view the financial transaction amount in the division's base currency in the **Financial Transaction** screen whenever additional FT GL entry was created for the financial transaction in the **CI_FT_GL_EXT** table. Now, depending on whether additional FT GL entry is created for the financial transaction in the division's base currency or division's local currency, you can view the financial transaction amount in the division's base currency or division's local currency, respectively, in the **Financial Transaction** screen.

To implement this feature, the following algorithm types are enhanced:

- FTFREZGLEXTN
- C1-GLCE

Now, these algorithm types provide the ability to create additional FT GL entries for a financial transaction in the GL's reporting currency (i.e. division's local currency).

Financial Transaction

Payment segments, adjustments and bill segment have a corresponding financial transaction. The financial transaction is created by the system when any of the source transactions is created. It contains the financial effects of its corresponding adjustment / bill segment / payment segment.

Fastpath: For background information, refer to The Big Picture Of Financial Transactions.

The topics in the section describe the financial transaction page.

Financial Transaction - Main

Use **Accounting and Receivables, Financial Transaction** to update and maintain financial transaction information.

Note: Navigating from source transactions. On the bill, payment, and adjustment maintenance pages, you can click the financial transaction go to button to transfer to this page.

Description of Page

Most of the attributes on this page are display-only. The following points describe the conditions under which certain fields may be modified:

- **Accounting Date** may be modified until the financial transaction (FT) is interfaced to the GL (i.e., until the **GL Distribution Status** is `Distributed`).
- **New Charge** may be modified until the FT is swept onto a bill (when the next bill is completed for the FT's account). If **New Charge** is modifiable and turned off, **Arrears Date** must be specified.
- **Show on Bill** and **Correction** can be modified until the FT is frozen.

The remainder of this section defines each of the fields on the page.

FT Type is the type of financial transaction. Values are: `Adjustment`, `Adjustment Cancellation`, `Bill Segment`, `Bill Segment Cancellation`, `Pay Segment`, and `Pay Segment Cancellation`. Click the go to button to view the originating transaction on the adjustment, bill, or payment page.

Copyright © 2024, Oracle and/or its affiliates.

FT ID is the system-assigned, unique identifier of the financial transaction.

Contract ID is the system-assigned, unique identifier of the contract to which the financial transaction is linked.

Bill ID is the system-assigned, unique identifier of the bill to which the financial transaction is linked. This field is only populated after the FT is swept onto a bill (and this happens when the next bill is completed for the FT's account).

Sibling ID is the system-assigned, unique identifier of the source transaction associated with the FT.

- If this FT is associated with a bill segment, **Sibling ID** contains the unique identifier of the bill segment.
- If this FT is associated with a payment segment, **Sibling ID** contains the unique identifier of the payment segment.
- If this FT is associated with an adjustment, **Sibling ID** contains the unique identifier of the adjustment.

Parent ID contains the following:

- If this FT is associated with a bill segment, **Parent ID** contains the unique identifier of the bill with which the bill segment is associated.
- If this FT is associated with a payment segment, **Parent ID** contains the unique identifier of the payment with which the payment segment is associated.
- If this FT is associated with an adjustment, **Parent ID** contains the adjustment type.

Create Date/Time are the date and time the FT was created.

Accounting Date is the accounting date that will be used by the general ledger to define the accounting period(s) into which the FT will be booked.

Division is the division associated with the FT. This comes from the division linked to the FT's contract's contract type.

GL Division is the GL division associated with the FT. This comes from the GL division linked to the FT's contract's contract type.

Show on Bill indicates if the FT will be shown on the customer's bill. You should only turn this off for erroneous FTs that should not be shown to the customer. For example, if you cancel / rebill a bill segment and you want to suppress the resultant FTs on the printed bill, turn this switch off.

New Charge indicates if the FT's charge only starts aging when the FT is swept onto the next bill produced for the account. If you turn this switch off, you must enter the date the FT starts aging in **Arrears Date**.

Not In Arrears indicates if the FT's financial impact has been canceled and therefore should not be considered for arrears purposes. This switch is turned on by the system when a FT's source transaction in canceled (both the original FT and the cancellation FT are marked as **Not In Arrears**).

Match Event ID is the system-assigned, unique identifier of the match event to which the financial transaction is linked. This field is only enabled for Open Item Accounts. Be aware that changing a financial transaction's match event can result in the balancing / unbalancing of the prior and newly referenced match events.

Arrears Date is the date the FT starts aging. This field is typically blank until the FT is swept onto the account's bill. If you want to start aging an FT on an historical date (for whatever reason), enter the appropriate **Arrears Date**. (Note - you must uncheck the "New Charge" box first in order to do this.)

Correction causes the FT to be summarized in the correction area of the bill-at-a-glance. This is set by the system automatically when a bill segment is cancelled / rebilled after its parent bill is completed (i.e., sent to the customer).

Redundant indicates if the FT's financial impact is considered irrelevant. This only happens after an FT has reached an age that is no longer relevant for aging purposes (e.g., when the FT is older than 120 days - or whatever age has been set as the Oldest Bucket Age on the Installation Options) and when its balance is exactly equal to zero with respect to other redundant FTs of the contract.

Presentment Bill ID indicates the bill on which the financial transaction is presented.

The **Frozen** switch is turned on when the FT has been frozen (i.e., posted to the contract's payoff and/or current balances). If this switch is on, both **Freeze Date/Time** and **Frozen By** are populated.

Current Amount contains the FT's impact on the contract's current amount.

Payoff Amount contains the FT's impact on the contract's payoff amount.

Copyright © 2024, Oracle and/or its affiliates.

Fastpath: For more information, refer to Current Balance versus Payoff Balance.

Currency Code is the currency code associated with the FT's account.

GL Distribution Status defines the status of the FT in respect of its interface to the general ledger. When an FT is first created, its status is `Pending`. When an FT is frozen, this value is set to `Generated`. If you modify the GL details, the status becomes `Modified`. After it has been marked for interface to the general ledger by the `GLS` process, the status becomes `Distributed`.

GL Extract Dates display the **Scheduled** date that the GL details will be marked for interface to the general ledger. Note that this date is typically only set to a future date for FTs associated with automatic payments. **Actual** displays the date that the GL details were actually posted to the general ledger.

The grid at the bottom shows the FT's debits and credits (i.e., the detail journal lines). Debits are shown as positive amounts, credits are negative amounts. The following points describe rules governing if and how the information in the grid can be modified:

- The FT GL details may not be modified if the FT is `Pending` or `Distributed`.
- All **Amounts** must sum to zero.
- Multiple FT GL details may be set to affect the **Total Amount** (although they are normally generated with one GL checked to affect total amount).
- All FT GL's checked to affect **Total Amount** must sum to the **Payoff Amount** of the FT.

The following information is displayed:

- **Seq** uence number is the system-assigned, unique identifier of the journal line.
- **Total Amount** defines if the journal line contains the total of the other journal lines. For example, on a bill segment for financial service, this would be turned on for the receivable account because its amount equals the sum of the other payable and revenue GL accounts.
- **Distribution Code** is the CIS distribution code from which the GL account constituents are derived.
- If the **GL Account** has been populated, the GL account number is displayed.

Fastpath: Refer to GLASSIGN - Assign GL Account Numbers To GL Details for more information about how the GL account is populated.

- **Amount** defines the journal line's amount.
- **Statistic Amount** defines the statistical amount that will be posted to the GL. This value is only populated on distribution lines created for rate components designated as affecting GL statistical quantity.
- **Characteristic Type** and **Characteristic Value** describe the characteristic value that was used when the line's amount was calculated. This information is only displayed if the journal line was derived from bill calculation lines that were calculated using a bill factor (because only bill factor's use characteristic values). Refer to *An Illustration Of A Bill Factor And Its Characteristics* for more information.

Note: Tax reporting. It is important to note that a journal line's characteristic value is NOT interfaced to the GL. We have included this attribute on the journal line so that tax reporting can be performed from this system (tax reporting typically necessitates showing each taxing authority - the characteristic value - that participated in a given tax payable GL account).

- If you have configured your installation options to indicate that fund accounting is `practiced`, the description of the **Fund** associated with this distribution code is displayed.

Financial Transaction - FT Process

Use **Accounting and Receivables, Financial Transaction, FT Process** to view those ancillary processes that may be triggered as a result of this financial transaction.

Copyright © 2024, Oracle and/or its affiliates.

Note: Algorithms cause financial transactions to be linked to batch processes. Financial transactions get associated with a given process / batch number when certain algorithms are executed. For example, when a bill segment is frozen, the system executes the freeze algorithms associated with the bill segment's contract's contract type. One of these algorithms may check if there are service providers who are interested in the bill segment's product/charge type consumption, and if so, schedule the product/charge type consumption to be downloaded by linking the bill segment's financial transaction to the service provider's product/charge type consumption download process.

Description of Page

This page shows the **Batch Processes** associated with a given **Financial Transaction**. Information on this page may not be modified. This information appears purely for audit purposes.

Account Financial History

This page shows how an account's current and payoff balance have changed over time. Use **Report and Inquiry, Account Financial History** to open this page.

Description of Page

This page is dedicated to a grid that shows an account's financial events. These events are grouped together by Arrears Date, Financial Transaction Type and Parent Id (therefore if there were two payments on the same date, two rows would appear).

Note: Multiple adjustments of the same type on the same date. If multiple adjustments with the same adjustment type exist on the same date, their total amount will appear as a single row on this query.

You can use this grid to both view high-level information about these events and to transfer to the respective page in which an object is maintained. The following columns are displayed in the grid:

Arrears Date This is the date the event starts aging. This column will be blank if the FT has not started aging yet.

Financial Transaction Type This column indicates the type of financial event: `Bill Segment`, `Pay Segment`, `Bill Segment Cancellation`, `Pay Segment Cancellation`, `Adjustment and Adjustment (Cancel)`. If the event is related to an adjustment, the adjustment type's description is displayed instead of "Adjustment".

Bill/Payment/Adjustment ID This column indicates the bill, payment, or adjustment for which the financial transaction (FT) is created.

Current Amount This column shows the financial event's effect on the account's current balance.

Current Balance This column shows the account's current balance after the financial event. Note that if the account has financial transactions in multiple currencies, zero is displayed in this column.

Payoff Amount This column shows the financial event's effect on the account's payoff balance. The value is grayed out if it is the same as the current amount.

Payoff Balance This column shows the account's payoff balance after the financial event. The value is grayed out if it is the same as the current balance. Note that if the account has financial transactions in multiple currencies, zero is displayed in this column.

If you need to see more information about a specific financial transaction, click the go to button to transfer to the respective page in which the information is maintained.

Fastpath: For information about current and payoff balance, refer to Current Amount versus Payoff Amount.

Copyright © 2024, Oracle and/or its affiliates.

Account Bill/Payment History

This page shows an account's bills and payments. Use **Report and Inquiry, Account Bill/Payment History** to open this page.

CAUTION: For balance-forward accounts, bill rows contain the balance presented on the respective bill, and payment rows contain the amount of the respective payment. However, for open-item accounts, this query behaves differently - see the description of page below for the details.

Description of Page

This page is dedicated to a grid that shows the account's bills, credit notes, correction notes, payments and payment cancellations. You can use this grid to both view high-level information about these objects and to transfer to the respective page on which an object is maintained.

The area beneath **Account ID** provides you with options that control which transactions appear in the grid. The following points describe the various options:

- Use **Transaction Type Filter** to restrict the type of transactions that appear in the grid. The following options are available:
  - `All`. This option shows all transactions.
  - `Bill`. This option shows all bills.
  - `Not Billed Yet`. This option shows a single line with a summary of frozen financial transactions that have not appeared on a bill yet.
  - `Payment`. This option shows all payments.
  - `Payment Cancellation`. This option shows all payment cancellations.
- Use **Match Event Status Filter** to restrict the transactions based on the status of their match event. This filter only appears if the bill's account is an open-item customer. The following options are available:
  - `All`. This option shows all transactions regardless of the status of their match events.
  - `Balanced`. This option shows all transactions with at least one `balanced` match event.
  - `Disputed`. This option shows all transactions with at least one `disputed` match event.
  - `Unbalanced`. This option shows all transactions with at least one `unbalanced` match event.
  - `Unmatched`. This option shows all transactions with at least one financial transaction that is not linked to a match event.
- Use **Date Range From** and **To** to restrict the transactions based on arrears date.

CAUTION: Don't forget to click the search button after changing the filters or after selecting a new Account ID.

For balance-forward accounts, bill rows contain bill information including the balance presented on the respective bill, and payment rows contain the amount of the respective payment.

For open-item accounts, the grid behaves differently:

- The amount on bill rows is equal to the sum of the current charges, adjustments and corrections on the bill. Payment rows contain the amount of the respective payment.
- It indicates the bill or payment for which the financial transaction is created.
- Each row contains an indication if all of its financial transactions are fully matched.
- A summary of the match status of its financial transactions is shown in the adjacent columns:
  - **Balanced** contains the count and total amount of financial transactions linked to this activity that are linked to `balanced` match events.
  - **Unbalanced** contains the count and total amount of financial transactions linked to this activity that are linked to `unbalanced` match events.
  - **Disputed** contains the count and total amount of financial transactions linked to this activity that are linked to `disputed` match events.

Copyright © 2024, Oracle and/or its affiliates.

- **Unmatched** contains the count and total amount of financial transactions linked to this activity that are not linked to any match event.
- You can use the hyperlinks to view the detailed financial transactions that are summarized in each cell.

Contract Financial History

This page is dedicated to a grid that shows the financial transactions linked to a contract. Use **Report and Inquiry, Contract Financial History** to open this page.

Description of Page

This page is dedicated to a grid that shows a contract's financial transactions (FT). You can use this grid to both view high level information about these objects and to transfer to the respective page in which an object is maintained.

The following columns are displayed in the grid:

Arrears Date This is the date the FT starts aging. This column will be blank if the FT has not started aging yet.

Financial Transaction Type This column indicates the type of financial event: `Bill Segment`, `Pay Segment`, `Bill Cancellation`, `Pay Cancellation`, `Adjustment` and `Adjustment (Cancel)`. If the event is related to an adjustment, the adjustment type's description is displayed instead of "Adjustment".

Current Amount This column shows the FT's effect on the contract's current balance.

Current Balance This column shows the contract's current balance after the financial event.

Payoff Amount This column shows the FT's effect on the contract's payoff balance. The value is grayed out if it is the same as the current amount.

Payoff Balance This column shows the contract's payoff balance after the financial event. The value is grayed out if it is the same as the current balance.

If you need to see more information about a specific financial transaction, click the go to button to transfer to the respective page in which the information is maintained.

Fastpath: For information about current and payoff balance, refer to Current Amount versus Payoff Amount.

Contract Cash Accounting Balance

Note: Only relevant if you practice cash accounting. This transaction is only relevant if you practice cash accounting (e.g., you only pay the taxing authorities when the customer pays you). Refer to Payables Cash Accounting for more information about cash accounting.

This page displays a grid that shows a contract's cash accounting balance for each cash holding distribution code that's been booked to. Use **Accounting and Receivables, Contract Cash Accounting Balance** to open this page.

Description of Page

The following columns are displayed in the grid:

Cash Accounting Distrib Code This is a general ledger distribution code used as the holding account for cash accounting. Each holding account that's been used by this contract will appear here.

Payable Distribution Code This is the general ledger distribution code to which the payable is (or will be) transferred when the cash event occurs.

Fastpath: For more information on how the distribution codes are set up, refer to Setting Up Distribution Codes.

Copyright © 2024, Oracle and/or its affiliates.

Payoff Balance This column shows the contract's balance for each cash holding account. If this amount is non-zero, it indicates that for this contract we are still holding some payables back until payment is received by the customer, at which time some or all of this amount will be transferred to the payable distribution code.

Fastpath: For information about cash accounting, refer to Payables Cash Accounting.

Balance Control

This page is used to summarize the financial transactions that belong to a particular balance control group.

Fastpath: Refer to The Big Picture of Balance Control **for more information**.

Note: Balance Control information is current only as of the Create Date/Time listed. The information displayed on the page is captured when the balance control background process is run. Any financial activity subsequent to the create date and time will not be shown.

Use **Accounting and Receivable, Balance Control** to open this page.

Description of Page

This page displays summarized financial information for a particular balance control group. It also details financial information for all contract types that belong to the group.

The following fields are displayed on the page:

Group ID This is a sequential value that uniquely identifies a particular balance control run. The Group ID is assigned and incremented automatically every time the Balance Control background process is run.

Status This is the status of the balance control record: `Pending`, and `Complete`. The status of the balance control record will only be pending while the balance control background process is being executed. Balance control information is only reliable if the status is complete.

Create Date/Time This is the date and time that the balance control record was created. Financial information displayed on this page will only contain information for financial transactions frozen before this date and time.

Current Amount This is the sum of the current amounts of all FTs that belong to the balance control group.

Payoff Amount This is the sum of the payoff amounts of all FTs that belong to the balance control group.

Current Balance This is the sum of the current amounts of all FTs that belong to this balance control group plus all earlier balance control groups. Therefore this is the instantaneous current balance of the entire CIS as of the create date and time.

Payoff Balance This is the sum of the payoff amounts of all FTs that belong to this balance control group plus all earlier balance control groups. Therefore this is the instantaneous payoff balance of the entire CIS as of the create date and time.

The number of FT information details the number of bill segments, pay segments, adjustments and their cancellations that belong to the balance control group.

Adjustment The total number of adjustments that belong to this balance control group.

Adjustment Cancellation The total number of adjustment cancellations that belong to this balance control group.

Bill Segment The total number of bill segments that belong to this balance control group.

Bill Cancellation The total number of bill cancellations that belong to this balance control group.

Pay Segment The total number of pay segments that belong to this balance control group.

Pay Cancellation The total number of pay cancellations that belong to this balance control group.

The **Contract Type** scroll summarizes the FTs that belong to a particular contract type within the balance control group - otherwise all fields are defined as above. You can scroll through all contract types that belong to the balance control group.

Copyright © 2024, Oracle and/or its affiliates.

Contract Type The contract type being summarized.

Match Event

CAUTION: Match Events are only used if you practice Open Item Accounting.

Match events are used to match debit and credit financial transactions together. When financial transactions are linked to a `balanced` match event, they no longer affect the customer's arrears.

You can use the match event page to do the following:

- Add, change, delete, and cancel match events.
- Add and remove financial transactions from / to a match event.
- Designate financial transactions as being "in dispute" (disputed financial transactions do not affect aged debt until they are resolved).
- View all unmatched financial transactions linked to an account.

Fastpath: Refer to Match Events for a full description of the lifecycle of a match event and a description of how the system automatically creates match events.

Match Event - Main

A typical match event matches a bill's financial transactions with a payment's financial transactions. The **Main** page allows you to define the bill(s) and payment(s) whose financial transactions are matched together. Use **Payments, Match Event** to open this page.

Note: Debit must equal credits. Before a match event impacts a customer's arrearage, its debits and credits must net to zero for every contract referenced on the match event. Until that time, the financial transactions on a match event continue to affect arrearage. The only exception is the case of disputes.

Note: You can match any debit and credit. While the above describes the matching of bills and payments, it's important to remember that a match event matches any type of debit with any type of credit. This means that a match event could match a bill with a credit or correction note, or a payment with a payment cancellation.

Note: You can match any number of payments under a match event. While most match events deal with a single bill and a single payment, there's no limitation to the number of payments on a match event. The only restriction is that the debits and credits must net to zero for all contracts.

Note: It is better not to mix multiple bills on a single match event. For purposes of bill balance information, it is strongly recommended that you compose your match events with financial transactions limited to a single bill. If you mix financial transactions from multiple bills on a single match event you will not be able to determine the unpaid balance of a partially paid bill.

Note: You can match specific financial transactions. While most match events deal with every financial transaction on a bill and payment, a match event can deal with individual financial transactions. For example, a match event could match a bill segment with a combination of a payment segment and a write-off adjustment. If you need to add or remove a specific financial transaction (i.e., a bill segment, payment segment, or adjustment), navigate to the **FT Details** tab. Perhaps a better way to differentiate between this page and the **FT Details** tab is to consider the example of a bill with 100 bill segments. When you link this bill to a match event, you are actually linking its 100 financial transactions. If you wanted to add only a subset of this bill's financial transactions to the match event, you'd use the **FT Details** tab.

Fastpath: Refer to How To for instructions describing how to perform common match event maintenance functions.

Copyright © 2024, Oracle and/or its affiliates.

Description of Page

This page is used to maintain the financial transactions (FTs) that are linked to a match event. The remainder of this section defines each of the fields on the page.

Match Event Info and **Match Event ID** only appear after the match event exists on the database. The **ID** is a system assigned random number that stays with the match event for life. **Match Event Info** is a concatenation of important details about the match event and its account.

The next section contains the sum of the **Debit** and **Credit** financial transactions linked to the match event. If the debits and credits do not sum to zero, the **Difference** is also shown.

Account ID defines the account whose financial transactions are matched under this match event. This field is gray after the match event is added to the database.

Match Event Status defines the state of the match event.

- Match events are initially created in the `open` state. Please note that you may delete an `open` match event.
- The system automatically changes an `open` event's status to `balanced` when the sum of the **Debit(s)** equals the sum of the **Credit(s)** *for each contract on the match event*. It's worth stressing that a match event may contain financial transactions from many contracts and each contract's financial transactions must sum to zero before the match event can become `balanced`.
- You may re `open` a `balanced` event (by adding / removing items so that the match event becomes unbalanced).
- You can cancel a `balanced` or `open` match event by changing the **Match Event Status** to `cancelled`. You must also define a **Cancel Reason** if you cancel a match event. Refer to How Are Match Events Cancelled? for more information about cancellation.

Turn on **Dispute** if this match event exists to designate certain financial transactions as disputed. In addition,

- Describe the reason for the dispute in **Remarks**.
- Link the disputed financial transactions to the match event by selecting the bill(s) below. If only a subset of a bill is disputed:
-
  - Click on the respective hyperlink in the **Unmatched Debits** column grid (this will transfer you to the next tab with these financial transactions displayed).
  - Select the bill segments that you want to designate as disputed.
  - Press the **Link / Unlink** button to link the selected bill segments to the match event.

> **Note: Disputing items on a balanced event.** The **Dispute** switch will be protected when the match event is `Balanced` or `Cancelled`. If the items on a `balanced` event are being disputed, you must add / remove items so that the match event becomes `Open` before you can turn on the **Dispute** switch.

The remainder of the page is dynamic depending on the **Match Event Status**:

- If the status is `Balanced` or `Open` , a grid appears containing a summary of the bills, payments, credit or correction notes, and payment cancellations that contribute financial transactions to the match event (note, we refer to these collectively as "contributing objects" in the following discussion). The following columns appear in this grid:
-
  - **Transaction Type** defines whether the contributing object is a `Bill` , `Payment` , `Payment Cancellation` or `Credit` or `Correction Note`. In the rare situation when unbilled financial transactions are linked to the match event, a transaction type of `Not Billed Yet` appears. If the contributing object is a bill its bill id is also displayed. If sequential bill functionality is enabled, the bill's sequential id is displayed instead.
  - **Matched Activity Information** contains summary information about the contributing object. This column is blank if the transaction type is `Not Billed Yet`.
  - The remaining columns contain the count and amount of each contributing object's financial transactions categorized as follows (note, you can drill down on the count or amount to see the specific financial transactions (FT's) on the next tab).
- **Matched Debits** summarizes the contributing object's debit FT's that are linked to this match event. A checkbox appears if the count is greater than zero. If you select the checkbox and press the **Link / Unlink** button, these debits will be removed from the match event.

Copyright © 2024, Oracle and/or its affiliates.

- **Matched Credits** summarizes the contributing object's credit FT's that are linked to this match event. A checkbox appears if the count is greater than zero. If you select the checkbox and press the **Link / Unlink** button, these credits will be removed from the match event.
- **Other Debits** summarizes the contributing object's debit FT's that are NOT linked to this match event.
- **Other Credits** summarizes the contributing object's credit FT's that are NOT linked to this match event.

Note: Adding more financial transactions to a balanced match event. When a match event is `Balanced` , the grid showing unmatched FT's is suppressed (and therefore you can't add additional FT's to the match event). To expose this grid, simply change the status of the match event to `Open`.

- If the status is `Open` , a second grid appears containing a summary of the bills, payments, credit notes, correction notes and payment cancellations with at least one unmatched financial transaction (we refer to these collectively as "unmatched objects" in the following discussion). The following columns appear in this grid:
- - **Transaction Type** defines whether the unmatched object is a `Bill` , `Payment` , `Payment Cancellation, Credit` or `Correction Note`. A transaction type of `Not Billed Yet` is used for unbilled financial transactions (e.g., adjustments generated between bills). If the unmatched object is a bill its bill id is also displayed. If sequential bill functionality is enabled, the bill's sequential id is displayed instead.
  - **Unmatched Activity Information** contains summary information about the unmatched object. This column is blank if the transaction type is `Not Billed Yet`.
  - The remaining columns contain the count and amount of each unmatched object's financial transactions categorized as follows (note, you can drill down on the count or amount to see the specific financial transactions (FT's) on the next tab).
- **Unmatched Debits** summarizes the unmatched object's debit FT's that are not linked to any match event. A checkbox appears if the count is greater than zero. If you select the checkbox and press the **Link / Unlink** button, these debits will be added to the match event.
- **Unmatched Credits** summarizes the unmatched object's credit FT's that are not linked to any match event. A checkbox appears if the count is greater than zero. If you select the checkbox and press the **Link / Unlink** button, these credits will be added to the match event.
- **Matched Debits** summarizes the unmatched object's debit FT's that are linked to any match event.
- **Matched Credits** summarizes the unmatched object's credit FT's that are linked to any match event.

Note: Credit Notes or Correction Notes. The Bill Correction option on the Installation table controls whether Credit Notes or Correction Notes are allowed. The default Transaction Type displayed is for Credit Note. If your implementation uses Correction Notes, you'll need to change the transaction type filter to reflect this. This is done by overriding the label for lookup value `CRNT` on the customizable lookup field `TXN_FLTR_TYPE_FLG`.

The last section contains the running total of the **Debit** and **Credit** financial transactions that have been selected / unselected in the above grids. If debits and credits do not sum to zero, the **Difference** is also shown. These values differ from the values on the top of the page as they are updated when a user selects / unselects an object (whereas the values at the top of the page are only updated when the database is changed).

The **Link / Unlink** button becomes enabled when you select an object in the grids. When you press this button, the financial transactions related to the object are added to / removed from the match event.

Match Event - FT Details

On the **Main** tab, you can add and remove bills, payments, credit notes, correction notes and payment cancellations to / from a match event. When you do this, you are actually adding and removing *all* of the financial transactions linked to these objects. For example, when you link a bill with 100 bill segments to a match event, you are actually linking its 100 financial transactions.

You need only use the **FT Details** tab when you need to add or remove specific financial transactions. For example, you would use the **FT Details** tab if you need to remove 1 of a bill's 100 financial transactions from a match event.

Use **Payments, Match Event, FT Details** to open this page (note, you can also open this page using many of the hyperlinks on the Main and Contract Subtotals tabs).

Copyright © 2024, Oracle and/or its affiliates.

Description of Page

This page is used to maintain the financial transactions (FTs) that are linked to a match event. The remainder of this section defines each of the fields on the page.

Match Event Info and **Match Event ID** only appear after the match event exists on the database. The **ID** is a system assigned random number that stays with a match event for life. The **Match Event Info** is a concatenation of important details about the match event and its account.

The next section contains the sum of the **Debit** and **Credit** financial transactions linked to the match event. If the debits and credits do not sum to zero, the **Difference** is also shown.

The following **Filters** work together to restrict the financial transactions that appear in the grid. The following points describe the various options (note, don't forget to press the search button after specifying the various filter options):

- Use **Transaction Type Filter** to restrict the type of transactions that appear in the grid. The following options are available:
- - `All`. Use this option if you do not wish to restrict financial transactions based on their transaction type.
  - `Bill`. This option allows you to view a specific bill's financial transactions. When this option is selected, an input field appears in which you identify the **Bill ID**.
  - `Credit Note`. This option allows you to view a specific credit note's financial transactions. When this option is selected, an input field appears in which you identify the **Bill ID** (every credit note has a unique bill ID).
  - `Correction Note`. This option allows you to view a specific correction note's financial transactions. When this option is selected, an input field appears in which you identify the **Bill ID** (every correction note has a unique bill ID).
  - `Not Billed Yet`. This option allows you to view financial transactions that haven't appeared on a bill yet (e.g., adjustments and corrections).
  - `Payment`. This option allows you to view a specific payment's financial transactions. When this option is selected, an input field appears in which you identify the **Payment ID**.
  - `Payment Cancellations`. This option allows you to view a specific canceled payment's financial transactions. When this option is selected, an input field appears in which you identify the **Payment ID**.

Note: Credit Notes or Correction Notes. The Bill Correction option on the Installation table controls whether Credit Notes or Correction Notes are allowed. The default Transaction Type displayed is for Credit Note. If your implementation uses Correction Notes, you'll need to change the transaction type filter to reflect this. This is done by overriding the label for lookup value `CRNT` on the customizable lookup field `TXN_FLTR_TYPE_FLG`.

- Use **Linkage Filter** to restrict the transactions based on the match event to which they are linked. The following options are available:
- - `All`. This option shows all financial transactions regardless of their match event.
  - `Linked to another Match Event`. This option shows financial transactions linked to a different match event.
  - `Linked to any Match Event`. This option shows financial transactions linked to any match event.
  - `Linked to this Match Event`. This option shows financial transactions linked to this match event.
  - `Unmatched`. This option shows financial transactions that are not linked to a match event.
- This filter is protected and set to `Linked to this Match Event` if the **Transaction Type Filter** is `All`.
- Use **Debit / Credit Filter** to restrict the transactions based on whether they are debits or credits. The following options are available:
- - `All`. This option shows all debit and credit financial transactions.
  - `Credit`. This option shows all credit financial transactions.
  - `Debit`. This option shows all debit financial transactions.
- Use the **Contract Filter** to define the types of contracts whose financial transactions appear in the grid. The following options are available:
- - `All.` Use this option if you do not wish to restrict financial transactions based on contract attributes.
  - `Contract ID.` Use this option to restrict financial transactions to those linked to a specific **contract**.

Copyright © 2024, Oracle and/or its affiliates.

- `Contract Type.` Use this option to restrict financial transactions to those whose contracts are linked to a given **Division** and **Contract Type**.

CAUTION: Don't forget to click the search button after changing the filters.

The **Select All**/ **Unselect All** buttons are used to select financial transactions to add to / remove from the match event (note, after selecting the desired financial transactions, you must also press the **Link / Unlink** button at the bottom of the page).

Note: 50 financial transactions at a time. Clicking **Select All** selects the first 50 bill segments in the grid. If more than 50 financial transactions exist, you must select them in batches.

The grid that follows contains the financial transactions (FT) that match your search criteria. The following information appears in the grid:

- Select box. Select the FT if you want to Link / Unlink it. You implicitly link FT's that are NOT already linked and you implicitly unlink FT's that are already linked. This field is protected if the FT is linked to another match event.
- **FT Amount.** This column contains the amount of the financial transaction.
- **FT Type.** This column indicates the type of financial transaction: `Bill Segment`, `Bill Segment Cancellation`, `Pay Segment`, `Pay Segment Cancellation`, `Adjustment and Adjustment Cancellation`.
- **Arrears Date.** This column contains the financial transaction's arrears date.
- **Contract Information.** This column contains a summary of the financial transaction's contract.
- **Remarks.** This column highlights the financial transaction's match status: `Linked to this match event`, `Not linked to a match event`, `Linked to another match event`- match event status.

The last section contains the running total of the **Debit** and **Credit** financial transactions that have been selected / unselected in the grid. If debits and credits do not sum to zero, the **Difference** is calculated. These values differ from the values on the top of the page as they are updated when a user selects / unselects an object (whereas the values at the top of the page are only updated with the database is changed).

The **Link / Unlink** button becomes enabled when you select a row in the grid. When you press this button, the financial transactions related to the object are added to / removed from the match event.

Match Event - Subtotals

Before a match event impacts a customer's arrearage, its debits and credits must net to zero for every contract referenced on the match event. This page shows the sum of the debits and credits for every contract that contributes at least one financial transaction to the match event. Use **Payments, Match Event, Subtotals** to open this page

Description of Page

This page shows the sum of the debits and credits for every contract that contributes at least one financial transaction to the match event.

Match Event Info and **Match Event ID** only appear after the match event exists on the database. The **ID** is a system assigned random number that stays with a match event for life. The **Match Event Info** is a concatenation of important details about the match event and its account.

The next section contains the sum of the **Debit** and **Credit** financial transactions linked to the match event. If the debits and credits do not sum to zero, the **Difference** is also shown.

The following **Filters** work together to restrict the contracts that appear in the grid. The following points describe the various options (note, don't forget to press the search button after specifying the various filter options):

- Use the **Contract Filter** to define the types of contracts whose financial transactions appear in the grid. The following options are available:
- - `All.` Use this option if you do not wish to restrict contracts.
  - `Contract ID.` Use this option to see a specific **contract**.

Copyright © 2024, Oracle and/or its affiliates.

  - `Contract Type.` Use this option to restrict contracts to those with a given **Division** and **Contract Type**.
- Use **Status Filter** to restrict the contracts based on whether the sum of the debits and credits they contribute to the match event. The following options are available:
  - `All`. Use this option if you do not wish to restrict contracts based on this status.
- `Balanced`. This option shows only contracts where the sum of debit and credits nets to zero on this match event.
  - `Unbalanced`. This option shows only contracts where the sum of debit and credits do not net to zero on this match event.

CAUTION: Don't forget to click the search button after changing the filters.

The grid that follows contains the contracts that match your search criteria. The following information appears in the grid:

- **Contract Information.** This column contains a summary of important information about the contract.
- **Difference.** This column contains the difference between the **Matched Debits** and **Match Credits**. If this is non-zero, the value appears in red.
- **Matched Debits.** This column contains the sum of debit financial transactions that this contract contributes to this match event.
- **Matched Credits.** This column contains the sum of credit financial transactions that this contract contributes to this match event.

How To Perform Common Match Event Functions

How To Find The Match Event Associated With A Financial Transaction

If you need to find the match event on which a financial transaction (FT) was matched, display the FT in question on Financial Transaction - Main and then use the "go to" button adjacent to the **Match Event** to drill to the match event.

Note: The easiest way to display a financial transaction is to find its corresponding bill, payment or adjustment and then drill down on the desired transaction.

How To Dispute An Item

Refer to Disputing Items for background information about disputes.

If a customer wants to dispute an item:

- Create a match event for the account.
- Turn the match event's dispute switch on.
- Link the disputed financial transaction to the match event.
- Describe in the match event's comments the reason for the dispute.

How To Match A Small Mismatch

Assume the following scenario arises:

- A bill is produced for $2000
- The customer pays $1993
- An unbalanced match event will result because the customer didn't pay exactly what is owed

If you want to match this payment to the bill (and leave $7 for the next bill), do the following:

- Create a transfer adjustment of $7 where the transfer from / to contract is the same. This results in a debit financial transaction (FT) of $7 and a credit FT of $7.
- Create a match event (or update the unbalanced match event) where the matched FTs are:

Copyright © 2024, Oracle and/or its affiliates.

  - The $1993 payment
  - The credit side of the transfer adjustment ($7)
- And the $2000 bill
- Then, if the customer pays their next bill in full, the $7 debit (associated with the transfer adjustment) will be swept onto it.

> **Note: Automating small mismatches.** The algorithm responsible for matching a payment to a specific bill can have a tolerance amount defined on it. If the payment is within the tolerance limit, this algorithm will do the above for you. In other words, you don't have to manually do the above if you populate the tolerance limit appropriately on this algorithm. Refer to DSOV BILL-ID for more information about this algorithm.

Billed Revenue

Oracle Revenue Management and Billing provides you with a facility to view the total adjustment and bill segment amount which is billed or cancelled in a particular duration. In other words, you can view information about different types of financial transactions which are frozen in a particular duration. The data is grouped based on the currency and distribution code. You can view the details of all adjustments or only those adjustments which are fetched on the bill.

Viewing Billed Revenue

Procedure

To view the billed revenue for a particular duration:

1. Click the **Menu** link in the **Application** toolbar.
   A list appears.
2. From the **Main** menu, select **Accounting and Receivable** and then click **Billed Revenue**.

   The **Billed Revenue** screen appears. It contains the following two sections:

   - **Search Criteria** - the **Search Criteria** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
From Date	Used to search for financial transactions which were frozen on or after the specified date. **Note:** The From Date cannot be later than the To Date.	Yes
To Date	Used to search for financial transactions which were frozen on or before the specified date. **Note:** The To Date cannot be earlier than the From Date.	Yes
All Adjustments	Used to indicate whether you want to view the details of all adjustments or only those adjustments which are fetched on the bill.	No

   - **Search Results** - On clicking the **Search** button, the search results appear based on the specified search criteria. The **Search Results** section contains the following columns:

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
Currency	Indicates the currency in which the financial transactions are created.
Distribution Code	Indicates the distribution code associated with the financial transactions.
Total Adjustment Amount	Displays the total amount of adjustment financial transactions which are frozen in the specified duration.
Total Adjustment Cancellation Amount	Displays the total amount of adjustment cancellation financial transactions which are frozen in the specified duration.
Total Bill Segment Amount	Displays the total amount of bill segment financial transactions which are frozen in the specified duration.
Total Bill Segment Cancellation Amount	Displays the total amount of bill cancellation financial transactions which are frozen in the specified duration.

3. Enter the search criteria in the **Search** zone.
4. Click **Search**.
   The information about different types of financial transactions which are frozen in the specified duration appears in the **Search Results** section.

Related Topics

For more information on...	See...
Billed Revenue screen	Billed Revenue on page 570

FT GL Drill Down Details

Until now, Oracle Revenue Management and Billing enabled you to extract the trial or actual FT GL records in a flat file using the **GL Download Extract (GLDL)** batch. This flat file was then used for uploading the data in the General Ledger (GL) system. However, there was no mechanism available through which you can do the following:

- Validate the extracts which are sent to the GL system
- Backtrack the FT GL records in ORMB

A new screen named **FT GL Drill Down Details** is introduced in this release. It contains the following zones:

- **Search GL Extract File** - Enables you to search for the GL extract files which are created through the GLDL batch. You can search for files which contain either actual FT GL records or trial FT GL records.
- **GL Extract File Details** - Enables you to view the list of GL records in the GL extract file.
- **Financial Transaction Details** - Enables you to view the list of financial transactions which are posted against the GL account in ORMB. If you are viewing the details of a GL extract file which contains actual FT GL records, then this zone lists the actual financial transactions posted against the GL account. However, if you are viewing the details of a GL extract file which contains trial FT GL records, then this zone lists the trial financial transactions posted against the GL account.
- **Bill Segment Details** - Enables you to view the information of the bill segment for which the financial transaction is created. If you are viewing the details of a GL extract file which contains actual FT GL records, then this zone lists the actual bill segment. However, if you are viewing the details of a GL extract file which contains trial FT GL records, then this zone lists the trial bill segment.
- **Adjustment Details** - Enables you to view the information of the adjustment for which the financial transaction is created.
- **Payment Details** - Enables you to view the information of the payment for which the financial transaction is created.

Copyright © 2024, Oracle and/or its affiliates.

To use this feature, you need to set the **G/L BATCH CODE** field in the **Financial Transaction** tab of the **Installation Options** screen to **GLDL**.

Search GL Extract File

The **Search GL Extract File** zone enables you to search for the GL extract records available in the system using various search criteria. It contains the following two sections:

- **Search Criteria** - The **Search Criteria** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Search By	Used to indicate whether you want to search for the trial or actual GL extract files available in the system. The valid values are: • **FT GL Details** - Used to indicate whether you want to search for an actual FT GL extract records. • **Trial FT GL Details** - Used to indicate whether you want to search for a trial FT GL extract details.	Yes (Conditional) **Note:** This field is required while searching for the trial or actual FT GL extract details.
Batch Control	Used to indicate whether you want to search for the GL extract files which are created as a result of executing a particular batch. **Note:** The **Search** (🔍) icon appears corresponding to this field. On clicking the **Search** icon, the **Batch Control Search** window appears where you can search for the respective batch control. At present you can only search FT GL extract records which are created by executing the **GL Download Extract** batch control.	Yes
Batch Number	Used to search GL extract files which belong to a particular batch number of the executed batch control.	No
Extract From Date	Used to search for GL extract files which are created by executing the **GL Download Extract** batch from a particular date onwards.	No
Extract To Date	Used to search for GL extract files which are created by executing the **GL Download Extract** batch till a particular date.	No

Note: You must specify at least one search criterion while searching for a GL extract file.

- **Search Results** - On clicking the **Search** button, the search results appear based on the specified search criteria. The **Search Results** section contains the following columns:

Column Name	Column Description
Batch Control	Displays the batch using which the GL extract file is created in the system.

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
	Note: It has a link. On clicking the link, the **Batch Control** screen appears where you can view the details of the respective batch control.
Batch Number	Displays the batch number which is created for extracting the GL record.
Amount	Displays the sum of the transaction amounts from all the GL records of the GL extract file.
Extraction Date Time	Displays the date and time when the GL extract file is created in the system.
File Name	Displays the file name using which the GL extract file is created in the system.

Note: Pagination is used to display limited number of records in this zone. You can use the navigation links, such as **Previous** and **Next** to navigate between pages.

On clicking the **Broadcast** (📶) icon corresponding to a GL extract file, the **GL Extract File Details** zone appears where you can view a list of GL records of the respective GL extract file.

Searching for a GL Extract File

Procedure

To search for a GL extract file:

1. Click the **Menu** link in the **Application** toolbar.
   A list appears.
2. From the **Main** menu, select **Accounting and Receivable** and then click **FT GL Drill Down Details**.

   The **FT GL Drill Down Details** screen appears.
3. Enter the required details in the **Search GL Extract File** zone.
4. Click **Search**.
   A list of GL extract files that meet the search criteria appears in the **Search Results** section.

Related Topics

For more information on...	See...
FT GL Drill Down Details screen	FT GL Drill Down Details on page 571

GL Extract File Details

The **GL Extract File Details** zone allows you to view the list of GL records available in the system for a particular GL extract file. It contains the following columns:

Column Name	Column Description
Batch Control	Displays the batch using which the GL extract file is created in the system.
	Note: It has a link. On clicking the link, the **Batch Control** screen appears where you can view the details of the respective batch control.
Batch Number	Displays the batch number which is created for extracting the GL record.

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
GL Division	Displays the GL division for which the GL record is created in the system.
	Note: The GL division is associated to the contract type using which the contract is created for the account in the system.
GL Account	Displays the account number for which the GL record is created in a GL division.
Amount	Displays the sum of the financial transactions generated for the corresponding GL division and GL account combination.
Accounting Period	Displays the period for which the GL record is created in the system.
Extraction Date Time	Displays the date and time when the GL extract file is created in the system.
File Name	Displays the file name using which the GL extract file is created in the system.

Note: Pagination is used to display limited number of records in this zone. You can use the navigation links, such as **Previous** and **Next** to navigate between pages.

You can filter the list using various criteria (such as, **GL Division**, **Currency**, **Extract From Date**, **Extract To Date**, **From Accounting Period**, **To Accounting Period**, and **GL Account**) available in the **Filter** area. By default, the **Filter** area is hidden. You can view the **Filter** area by clicking the **Filters** ( ) icon in the upper right corner of this zone.

By default, the **GL Extract File Details** zone does not appear in the **FT GL Drill Down Details** screen. It appears only when you click the **Broadcast** ( ) icon corresponding to a GL extract file in the **Search Results** section of the **Search GL Extract File** zone.

Financial Transaction Details

The **Financial Transaction Details** zone enables you to view the list of financial transactions that are posted against a particular GL record of the GL extract file. It contains the following sections:

- **Filter Criteria** - The **Filter Criteria** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Financial Transaction ID	Used to search for a particular financial transaction detail.	No
Financial Transaction Type	Used to search financial transactions which are created for a particular financial transaction type. The valid values are: • Adjustment • Adjustment Cancellation • Bill Cancellation • Bill Segment • Pay Cancellation • Pay Segment	No

- **Filter Results** - On clicking the **Search** button, the filter results appear based on the specified filter criteria. The **Filter Results** section contains the following columns:

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
Financial Transaction Information	Displays the concatenated string of information about the financial transaction separated by a comma (,).
	Note: It has a link. On clicking the link, the **Financial Transaction** screen appears where you can view the details of the respective financial transaction.
Financial Transaction Type	Displays the financial transaction type using which the financial transaction record is created in the system. The valid values are: • Adjustment • Adjustment Cancellation • Bill Cancellation • Bill Segment • Pay Cancellation • Pay Segment
Entity ID	Displays the financial transaction ID generated for the GL division.
Amount	Displays the financial transaction amount.
Distribution Code	Indicates the distribution code associated with the financial transaction.

Note: Pagination is used to display limited number of records in this zone. You can use the navigation links, such as **Previous** and **Next** to navigate between pages.

By default, the **Financial Transaction Details** zone does not appear in the **FTGL Drill Down Details** screen. It appears only when you click the **Broadcast** (📶) icon corresponding to a GL record in the **GL Extract File Details** zone.

On clicking the **Broadcast** (📶) icon corresponding to a financial transaction, the **Adjustment Details**, **Bill Segment Details**, or **Payment Details** zone appears depending upon the financial transaction type for which the financial transaction is created in the system.

Bill Segment Details

The **Bill Segment Details** zone enables you to view the information of the bill segment for which the financial transaction is created against a particular GL record of the GL extract file. It contains the following columns:

Column Name	Column Description
Bill Information	Displays the concatenated string of information about the bill for which the bill segment is created separated by a comma (,). In addition, a context menu appears corresponding to this field which helps in navigating to other screens in the application.
	Note: It has a link. On clicking the link, the **Bill** screen appears where you can view the details of the respective bill.
Bill Segment Information	Displays the concatenated string of information about the bill segment for which the financial transaction is created separated by a comma (,). In addition, a context menu appears corresponding to this field which helps in navigating to other screens in the application.

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
	Note: It has a link. On clicking the link, the **Bill Segment** screen appears where you can view the details of the respective bill segment.
Start Date	Displays the date from when the bill segment is considered for billing.
End Date	Displays the date till when the bill segment is considered for billing.
Amount	Displays the bill segment amount.
Price Item	Displays the price item for which the bill segment is created.

By default, the **Bill Segment Details** zone does not appear in the **FTGL Drill Down Details** screen. It appears only when you click the **Broadcast** (📶) icon corresponding to a financial transaction in the **Financial Transaction Details** zone.

Adjustment Details

The **Adjustment Details** zone enables you to view the information of the adjustment for which the financial transaction is created against a particular GL record of the GL extract file. It contains the following columns:

Column Name	Column Description
Adjustment Information	Displays the concatenate string of information about the adjustment for which the financial transaction is created separated by a comma (,).
	Note: It has a link. On clicking the link, the **Adjustment** screen appears where you can view the details of the respective adjustment.
Account ID	Indicates the account for which the adjustment is created.
Adjustment Type	Indicates the adjustment type using which the adjustment is created.
Amount	Displays the adjustment amount.

By default, the **Adjustment Details** zone does not appear in the **FTGL Drill Down Details** screen. It appears only when you click the **Broadcast** (📶) icon corresponding to a financial transaction in the **Financial Transaction Details** zone.

Payment Details

The **Payment Details** zone enables you to view the information of the payment for which the financial transaction is created against a particular GL record of the GL extract file. It contains the following columns:

Column Name	Column Description
Payment Information	Displays the concatenate string of information about the payment for which the financial transaction is created separated by a comma (,).
	Note: It has a link. On clicking the link, the **Payment** screen appears where you can view the details of the respective payment.
Account ID	Indicates the account for which the payment is created.
Amount	Displays the payment amount.

Copyright © 2024, Oracle and/or its affiliates.

By default, the **Payment Details** zone does not appear in the **FTGL Drill Down Details** screen. It appears only when you click the **Broadcast** (ᯤ) icon corresponding to a financial transaction in the **Financial Transaction Details** zone.

Viewing the Details of a GL Extract File

Procedure

To view the details of a GL extract file:

1. Search for the GL extract file in the **Search GL Extract File** zone.
2. In the **Search Results** section, click the **Broadcast** (ᯤ) icon corresponding to the GL extract file whose details you want to view.
   The **GL Extract File Details** zone appears.
3. View the details of the GL extract file in the **GL Extract File Details** zone.

Related Topics

For more information on...	See...
Search GL Extract File zone	Search GL Extract File on page 572
How to search for a GL extract file	Searching for a GL Extract File on page 573
GL Extract File Details zone	GL Extract File Details on page 573

Viewing Financial Transactions Posted in a GL Account

Procedure

To view the financial transactions posted in a GL account:

1. Ensure that the **GL Extract File Details** zone is expanded when you are viewing the GL extract file details.
2. Click the **Broadcast** (ᯤ) icon corresponding to the GL account whose financial transactions you want to view.
   The **Financial Transaction Details** zone appears.
3. View the financial transactions of the GL account in the **Financial Transaction Details** zone.

Related Topics

For more information on...	See...
GL Extract File Details zone	GL Extract File Details on page 573
Financial Transaction Details zone	Financial Transaction Details on page 574

Viewing the Financial Transaction Details

Procedure

To view the financial transaction details:

1. Ensure that the **Financial Transaction Details** zone is expanded when you are viewing the financial transactions posted in a GL account.

Copyright © 2024, Oracle and/or its affiliates.

2. Click the **Broadcast** (ᯤ) icon corresponding to the financial transaction whose details you want to view.

   The **Bill Segment Details** zone appears if the **Transaction Type** is a bill segment or bill cancellation.

   The **Adjustment Details** zone appears if the **Transaction Type** is an adjustment or adjustment cancellation.

   The **Payment Details** zone appears if the **Transaction Type** is a payment or payment cancellation.
3. View the financial transactions details in the **Bill Segment Details**, **Adjustment Details**, or the **Payment Details** zone.

Related Topics

For more information on...	**See...**
Financial Transaction Details zone	Financial Transaction Details on page 574
Bill Segment Details zone	Bill Segment Details on page 575
Adjustment Details zone	Adjustment Details on page 576
Payment Details zone	Payment Details on page 576

Chapter 7

Statements

Topics:

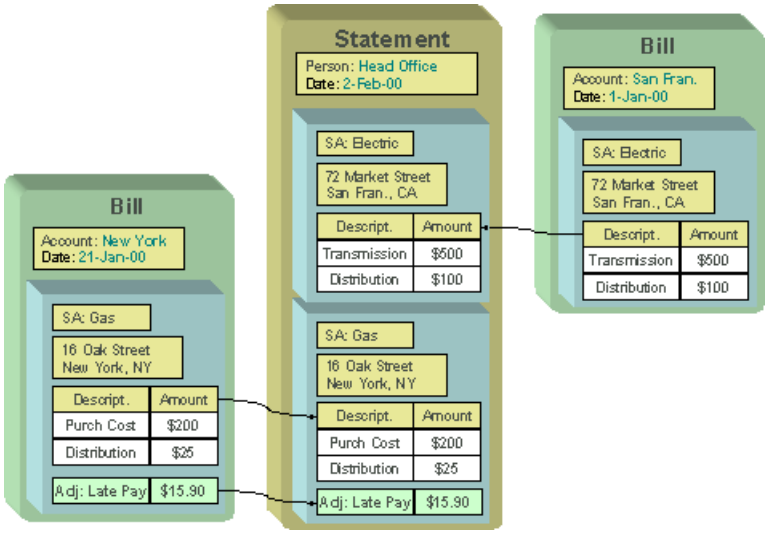
- The Big Picture of Complex Statements
- Statement Construct Maintenance
- Statement Maintenance

In this section, we describe how to manage your customer's statements.

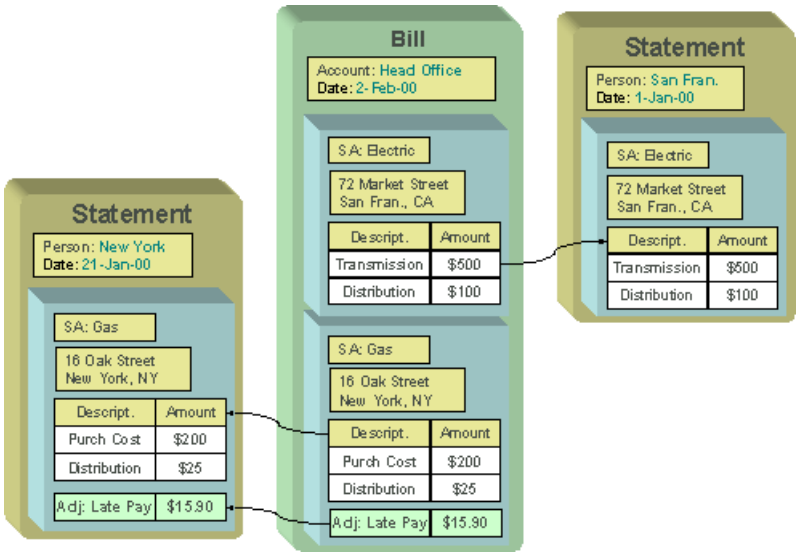
The Big Picture of Complex Statements

Statements allow you to set up a person to receive a consolidated report of the financial activity for one or more accounts and/or contracts. This allows you to model complex account scenarios, including the following examples.

- Satellite offices for a company are each responsible for paying their own bill, but the head office would like a consolidated report of the bills for all the satellite offices once a month. In this case, each satellite office is set up as an account covering the services for their office. The head office does not want to receive individual copies of each bill, but rather a consolidated report of the financial activity for all the offices.

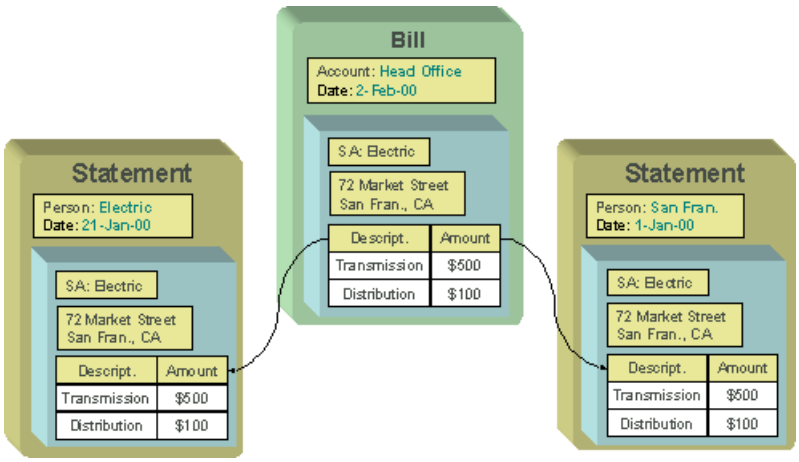


- The head office is responsible for paying the bill. The satellite offices should receive a copy of only their portion of the bill. In this case, a contract is set up for each satellite office under the head office's account. Each satellite office should receive a copy of the bill segment information for their office's contract.



- The head office is responsible for paying the bill. Various departments can refer the relevant sections of the bill. As illustrated by this scenario, a contract or account may appear on many statements.

Copyright © 2024, Oracle and/or its affiliates.



The topics in this section provide additional information about how to set up statements.

Constructing Statements

Statements are produced for a person. Each statement contains the financial transactions for one or more contracts. The contracts may belong to any number of accounts. Over time, a person may receive many statements.

When setting up a person to receive statements, it's important to note that a person can receive different statements for different groupings of contracts. For example, the head office may want a statement of the financial details for the western division offices separately from the eastern division offices.

For each separate statement a person wishes to receive, you will set up a statement construct. On the statement construct, you define the contracts and/or accounts whose financial information should appear on the statement. If you specify an account on a statement construct, billing information for all the contracts linked to the account will be included on the statement. The superset of contracts and accounts defined on a statement construct are called the "construct details".

Fastpath: Refer to Statement Construct Maintenance for more information.

Producing Statements

Statement Formats

A statement is the mechanism used by financial institutions to communicate with their customers regarding transactions and charges incurred by them. It is a periodic report that financial institutions issue to their customers showing their recent transactions, balance due and other key information. Unlike a bill, a statement is informative and you do not make payments against it. A statement is not legally binding. Statements are usually issued at the end of each billing cycle, which is typically about one month long. However, customers can choose the frequency at which they would like to receive the statements. A customer can also request for an adhoc statement.

Earlier, the financial institutions did not follow any standard format while generating the statements and each financial institution followed its own format. Though these statements were used to cater to the customer's needs, it was difficult for other financial institutions to comprehend and interpret the information in the statement, thereby resulting in confusions in inter-bank communications. To address this issue, standard statement formats were introduced that financial institutions were expected to follow.

Bank Services Billing (BSB) is an industry standard that governs the format of electronic bills sent out by financial institutions to large group customers (such as, corporations, governments, and institutions). BSB is a statement to report the corporate customers' usage of financial services and their related charges. It takes the form of a periodic electronic statement that lists all the chargeable service events that occurred during a reporting cycle, along with detailed tax and currency information. The standard was developed by Transaction Workflow Innovation Standards Team (TWIST). The

Copyright © 2024, Oracle and/or its affiliates.

primary aim of TWIST was to develop new and rationalize xml standards that connect the financial and physical supply chains, releasing the enormous value locked up in disjoined paper-based processes.

On the other hand, the Society for the Worldwide Interbank Financial Telecommunication (SWIFT) developed two main standards - MT (text-based) and MX (xml-based). To avoid the divergence caused by TWIST and SWIFT formats, ISO introduced a single standardization approach, which included a common development methodology, process and repository, called ISO 20022. The ISO 20022 standardization introduced the CAMT.086 format, which is a convergence of SWIFT and TWIST formats. The CAMT.086 BSB is the function equivalent of the TWIST BSB version 3.1. Nothing has been lost from the TWIST BSB; the only changes made were to conform to the ISO 20022 XML standards.

Oracle Revenue Management and Billing enables you to generate statements in the CAMT and TWIST formats. Two new statement route types - **CAMT** and **TWIST** - are shipped with the product. You can use these statement route types while defining a statement construct for a person. On executing the **Create Statements in CAMT or TWIST Format (CISTMEXT)** batch, the system will generate statements for all those statement constructs whose statement cycle's processing date which is earlier than the batch business date is in the **Pending** status. During the statement generation, the system will extract details from the bills in the CAMT or TWIST format which is specified in the statement construct.

You can also use the custom bill extraction process, if required. In such case, you can generate the statements using the legacy batch named **Create Statements (STMPRD)**.

Related Topic

For more information on...	See...
Statement Generation Process	Statement Generation Process on page 643
CAMT Format	CAMT Format on page 582
TWIST Format	TWIST Format on page 587
Field Mapping for the CAMT Format	Field Mapping for the CAMT Format on page 591
Field Mapping for the TWIST Format	Field Mapping for the TWIST Format on page 602
Tags Used in the CAMT Format	Tags Used in the CAMT Format on page 609
Tags Used in the TWIST Format	Tags Used in the TWIST Format on page 622
Sample Statement in the CAMT Format	Sample Statement in the CAMT Format on page 631
Sample Statement in the TWIST Format	Sample Statement in the TWIST Format on page 637

CAMT Format

The CAMT format captures the following information:

- **Header** - It consists of the header ID and pagination information.
- **Billing Statement Group** - The system groups the statements where the sender and receiver are the same. It contains information about the following:
  - **Sender** - A sender is a financial institution who sends the statement to the customer.
  - **Sender Individual Contact** - A sender individual contact is a person whom you can contact in case of technical problems at the sender's location.
  - **Receiver** - A receiver is a customer to whom you want to send the statement.
  - **Receiver Individual Contact** - A receiver individual contact is a person whom you can contact in case of technical problems at the receiver's location.
  - **Billing Statement** - It contains information from one or more regular bills for a statement cycle. There can be more than one billing statement section. However, these statements will be from the same sender and will be addressed to the same receiver. It contains information with respect to the following:
    - Accounts whose bill details are extracted in the statement
    - Branch to which the account belongs
    - Tax details of the account

Copyright © 2024, Oracle and/or its affiliates.

- Contact person at the branch office
- Exchange rates which are used for calculations
- Price items for which the charges are created
- Pricing used while creating the charges
- Distribution codes where tax is applicable
- Individual taxes applied on the charges
- Adjustments created with respect to the services and posted onto the bill

In addition, it contains the following:

- Sum total of charges where taxes are not applicable
- Sum total of charges where taxes are applicable
- Sum total of taxes calculated for each tax distribution code
- Sum total of all service charges and taxes billed to the receiver
- Sum total of all taxes charged to the receiver (i.e. customer)

The following code snippet illustrates how the tags are nested in the CAMT format:

```
<Document>
    <BkSvcsBllgStmt>
        <RptHdr>
            <RptId></RptId>
        </RptHdr>
        <MsgPgntn>
            <PgNb></PgNb>
            <LastPgInd></LastPgInd>
        </MsgPgntn>
        <BllgStmtGrp>
            <GrpId></GrpId>
            <Sndr>
                <Nm></Nm>
                <LglNm></LglNm>
                <PstlAdr>
                    <AdrTp></AdrTp>
                    <Dept></Dept>
                    <SubDept></SubDept>
                    <StrtNm></StrtNm>
                    <BldgNb></BldgNb>
                    <PstCd></PstCd>
                    <TwnNm></TwnNm>
                    <CtrySubDvsn></CtrySubDvsn>
                    <Ctry></Ctry>
                    <AdrLine></AdrLine>
                    <Flr></Flr>
                    <PstBx></PstBx>
                    <BldgNm></BldgNm>
                    <Room></Room>
                </PstlAdr>
                <Id>
                   <FIId>
                        <BICFI></BICFI>
                    </FIId>
                </Id>
                <CtryOfRes></CtryOfRes>
                <CtctDtls>
                    <NmPrfx></NmPrfx>
                    <Nm></Nm>
                    <PhneNb></PhneNb>
                    <MobNb></MobNb>
                    <FaxNb></FaxNb>
                    <EmailAdr></EmailAdr>
```

Copyright © 2024, Oracle and/or its affiliates.

```
                </CtctDtls>
            </Sndr>
            <SndrIndvCtct>
                <NmPrfx></NmPrfx>
                <Nm></Nm>
                <PhneNb></PhneNb>
                <MobNb></MobNb>
                <FaxNb></FaxNb>
                <EmailAdr></EmailAdr>
            </SndrIndvCtct>
            <Rcvr>
                <Nm></Nm>
                <LglNm></LglNm>
                <PstlAdr>
                    <AdrTp></AdrTp>
                    <Dept></Dept>
                    <SubDept></SubDept>
                    <StrtNm></StrtNm>
                    <BldgNb></BldgNb>
                    <PstCd></PstCd>
                    <TwnNm></TwnNm>
                    <CtrySubDvsn></CtrySubDvsn>
                    <Ctry></Ctry>
                    <AdrLine></AdrLine>
                    <Flr></Flr>
                    <PstBx></PstBx>
                    <BldgNm></BldgNm>
                    <Room></Room>
                </PstlAdr>
                <Id>
                    <OrgId>
                        <AnyBIC></AnyBIC>
                    </OrgId>
                    <FIId>
                        <BICFI></BICFI>
                    </FIId>
                </Id>
                <CtryOfRes></CtryOfRes>
                <CtctDtls>
                    <NmPrfx></NmPrfx>
                    <Nm></Nm>
                    <PhneNb></PhneNb>
                    <MobNb></MobNb>
                    <FaxNb></FaxNb>
                    <EmailAdr></EmailAdr>
                </CtctDtls>
            </Rcvr>
            <RcvrIndvCtct>
                <NmPrfx></NmPrfx>
                <Nm></Nm>
                <PhneNb></PhneNb>
                <MobNb></MobNb>
                <FaxNb></FaxNb>
                <EmailAdr></EmailAdr>
            </RcvrIndvCtct>
            <BllgStmt>
                <StmtId></StmtId>
                <FrToDt>
                    <FrDt></FrDt>
                    <ToDt></ToDt>
                </FrToDt>
                <CreDtTm></CreDtTm>
                <Sts></Sts>
                <AcctChrtcs>
```

Copyright © 2024, Oracle and/or its affiliates.

```
                    <AcctLvl></AcctLvl>
                    <CshAcct>
                     <Id>
                            <IBAN></IBAN>
                        </Id>
                        <Tp>
                            <Cd></Cd>
                            <Prtry></Prtry>
                        </Tp>
                        <Ccy></Ccy>
                    </CshAcct>
                    <AcctSvcr>
                        <FinInstnId>
                            <BICFI></BICFI>
                        </FinInstnId>
                        <Nm></Nm>
                        <PstlAdr>
                            <Dept></Dept>
                            <SubDept></SubDept>
                            <StrtNm></StrtNm>
                            <BldgNb></BldgNb>
                            <PstCd></PstCd>
                            <TwnNm></TwnNm>
                            <CtrySubDvsn></CtrySubDvsn>
                            <Ctry></Ctry>
                            <AdrLine></AdrLine>
                        </PstlAdr>
                    </AcctSvcr>
                    <CompstnMtd></CompstnMtd>
                    <DbtAcct>
                        <IBAN></IBAN>
                    </DbtAcct>
                    <DelydDbtDt></DelydDbtDt>
                    <SttlmAdvc></SttlmAdvc>
                    <AcctBalCcyCd></AcctBalCcyCd>
                    <SttlmCcyCd></SttlmCcyCd>
                    <HstCcyCd></HstCcyCd>
                    <Tax>
                        <ClctnMtd></ClctnMtd>
                        <Rgn></Rgn>
                    </Tax>
                    <AcctSvcrCtct>
                        <NmPrfx></NmPrfx>
                        <Nm></Nm>
                        <PhneNb></PhneNb>
                        <MobNb></MobNb>
                        <FaxNb></FaxNb>
                        <EmailAdr></EmailAdr>
                    </AcctSvcrCtct>
                </AcctChrtcs>
                <CcyXchg>
                    <SrcCcy></SrcCcy>
                    <TrgtCcy></TrgtCcy>
                    <XchgRate></XchgRate>
                    <UnitCcy></UnitCcy>
                    <QtnDt></QtnDt>
                </CcyXchg>
                <Svc>
                    <SvcDtl>
                        <BkSvc>
                            <Id></Id>
                            <Desc></Desc>
                            <SvcTp></SvcTp>
                        </BkSvc>
```

Copyright © 2024, Oracle and/or its affiliates.

```
                        <Vol></Vol>
                    </SvcDtl>
                    <Pric>
                        <Ccy></Ccy>
                        <UnitPric>
                            <Amt></Amt>
                            <Sgn></Sgn>
                        </UnitPric>
                        <Mtd></Mtd>
                    </Pric>
                    <PmtMtd></PmtMtd>
                    <OrgnlChrgSttlmAmt>
                        <Amt></Amt>
                        <Sgn></Sgn>
                    </OrgnlChrgSttlmAmt>
                </Svc>
                <TaxRgn>
                    <RgnNb></RgnNb>
                    <RgnNm></RgnNm>
                    <CstmrTaxId></CstmrTaxId>
                    <InvcNb></InvcNb>
                    <MtdC>
                        <SvcDtl>
                            <BkSvc>
                                <Id></Id>
                                <Desc></Desc>
                            </BkSvc>
                            <SvcChrgAmt>
                                <Amt></Amt>
                                <Sgn></Sgn>
                            </SvcChrgAmt>
                        </SvcDtl>
                        <TaxClctn>
                            <HstCcy></HstCcy>
                                <TaxblSvcChrgConvs>
                                    <SrcAmt>
                                        <Amt></Amt>
                                        <Sgn></Sgn>
                                    </SrcAmt>
                                    <HstAmt>
                                        <Amt></Amt>
                                        <Sgn></Sgn>
                                    </HstAmt>
                                </TaxblSvcChrgConvs>
                                <TtlTaxblSvcChrgHstAmt>
                                    <Amt></Amt>
                                    <Sgn></Sgn>
                                </TtlTaxblSvcChrgHstAmt>
                                <TaxId>
                                    <Nb></Nb>
                                    <Desc></Desc>
                                    <Rate></Rate>
                                    <TtlTaxAmt>
                                        <Amt></Amt>
                                        <Sgn></Sgn>
                                    </TtlTaxAmt>
                                </TaxId>
                                <TtlTax>
                                    <Amt></Amt>
                                    <Sgn></Sgn>
                                </TtlTax>
                        </TaxClctn>
                    </MtdC>
                    <SttlmAmt>
```

Copyright © 2024, Oracle and/or its affiliates.

```
                        <Amt></Amt>
                        <Sgn></Sgn>
                    </SttlmAmt>
                    <TaxDueToRgn>
                        <Amt></Amt>
                        <Sgn></Sgn>
                    </TaxDueToRgn>
                </TaxRgn>
                <SvcAdjstmnt>
                    <Tp></Tp>
                    <Desc></Desc>
                    <Amt>
                        <Amt></Amt>
                        <Sgn></Sgn>
                    </Amt>
                    <ErrDt></ErrDt>
                    <AdjstmntId></AdjstmntId>
                    <NewChrgAmt>
                        <Amt></Amt>
                        <Sgn></Sgn>
                    </NewChrgAmt>
                </SvcAdjstmnt>
            </BllgStmt>
        </BllgStmtGrp>
    </BkSvcsBllgStmt>
</Document>
```

Related Topics

For more information on...	See...
Tags Used in the CAMT Format	Tags Used in the CAMT Format on page 609
Field Mapping for the CAMT Format	Field Mapping for the CAMT Format on page 591
Sample Statement in the CAMT Format	Sample Statement in the CAMT Format on page 631

TWIST Format

The TWIST format captures the following information:

- **Header** - It contains the header ID.
- **Electronic Statement** - The system groups the statements where the sender and receiver are the same. It contains the following:
  - **Statement Header** - It contains information about the following:
    - **Sender** - A sender is a financial institution who sends the statement to the customer.
    - **Sender Individual Contact** - A sender individual contact is a person whom you can contact in case of technical problems at the sender's location.
    - **Receiver** - A receiver is a customer to whom you want to send the statement.
    - **Receiver Individual Contact** - A receiver individual contact is a person whom you can contact in case of technical problems at the receiver's location.
  - **Statement** - It contains information from one or more regular bills for a statement cycle. There can be more than one statement section. However, these statements will be from the same sender and will be addressed to the same receiver. It contains information with respect to the following:
    - Accounts whose bill details are extracted in the statement
    - Branch to which the account belongs
    - Tax details of the account
    - Contact person at the branch office
    - Exchange rates which are used for calculations

Copyright © 2024, Oracle and/or its affiliates.

- Price items for which the charges are created
- Pricing used while creating the charges
- Distribution codes where tax is applicable
- Individual taxes applied on the charges
- Adjustments created with respect to the services and posted onto the bill

In addition, it contains the following:

- Sum total of charges where taxes are not applicable
- Sum total of charges where taxes are applicable
- Sum total of taxes calculated for each tax distribution code
- Sum total of all service charges and taxes billed to the receiver
- Sum total of all taxes charged to the receiver (i.e. customer)

The following code snippet illustrates how the tags are nested in the TWIST format:

```
<Twist>
    <header></header>
    <electronicStatement>
        <statementHeader>
            <stmtSender>
                <name></name>
                <legalName></legalName>
                <contactInfo>
                    <postAddress>
                        <addressIdentifier></addressIdentifier>
                        <departmentName></departmentName>
                        <streetName></streetName>
                        <buildingNum></buildingNum>
                        <postCode></postCode>
                        <city></city>
                        <countrySubDivision>
                            <name></name>
                        </countrySubDivision>
                        <country></country>
                        <addressLine></addressLine>
                        <floor></floor>
                        <postBox></postBox>
                        <buildingName></buildingName>
                        <room></room>
                    </postAddress>
                </contactInfo>
                <orgId>
                    <orgIdNum></orgIdNum>
                </orgId>
                <contactInfo>
                    <individualContact>
                        <name></name>
                    </individualContact>
                    <phone></phone>
                    <fax></fax>
                    <Email></Email>
                </contactInfo>
            </stmtSender>
            <stmtReceiver>
                <name></name>
                <legalName></legalName>
                <contactInfo>
                    <postAddress>
                        <addressIdentifier></addressIdentifier>
                        <departmentName></departmentName>
                        <streetName></streetName>
```

Copyright © 2024, Oracle and/or its affiliates.

```
                    <buildingNum></buildingNum>
                    <postCode></postCode>
                    <city></city>
                    <countrySubDivision>
                        <name></name>
                    </countrySubDivision>
                    <country></country>
                    <addressLine></addressLine>
                    <floor></floor>
                    <postBox></postBox>
                    <buildingName></buildingName>
                    <room></room>
                </postAddress>
            </contactInfo>
            <orgId>
                <orgIdNum></orgIdNum>
            </orgId>
            <contactInfo>
                <individualContact>
                    <name></name>
                </individualContact>
                <phone></phone>
                <fax></fax>
                <Email></Email>
            </contactInfo>
        </stmtReceiver>
    </statementHeader>
    <statement>
        <account>
            <statementStartDate></statementStartDate>
            <statementEndDate></statementEndDate>
            <statementProductionDate></statementProductionDate>
            <statementStatus></statementStatus>
            <accountLevel></accountLevel>
            <iban></iban>
            <domicileBankIdentifier></domicileBankIdentifier>
            <compensationMethod></compensationMethod>
            <debitAccount></debitAccount>
            <delayedDebitDate></delayedDebitDate>
            <settlementAdvice></settlementAdvice>
            <accountBalanceCurrencyCode></accountBalanceCurrencyCode>
            <settlementCurrencyCode></settlementCurrencyCode>
            <hostCurrencyCode></hostCurrencyCode>
            <taxCalculationMethod></taxCalculationMethod>
            <taxRegionCode></taxRegionCode>
            <bankContactName></bankContactName>
            <bankContactPhone></bankContactPhone>
            <bankContactFax></bankContactFax>
            <bankContactEmail></bankContactEmail>
        </account>
        <currencyTranslation>
            <originalCurrency></originalCurrency>
            <targetCurrency></targetCurrency>
            <translationValue></translationValue>
            <basis></basis>
        </currencyTranslation>
        <service>
            <bankServiceID></bankServiceID>
            <serviceDescription></serviceDescription>
            <serviceType></serviceType>
            <volume></volume>
            <pricingCurrencyCode></pricingCurrencyCode>
            <unitPrice>
                <amount></amount>
```


Copyright © 2024, Oracle and/or its affiliates.

```
                    <currencyCode></currencyCode>
                </unitPrice>
                <priceMethod></priceMethod>
                <paymentMethod></paymentMethod>
                <originalChargeSettlement>
                    <amount></amount>
                    <currencyCode></currencyCode>
                </originalChargeSettlement>
            </service>
            <taxDetails>
                <taxRegion>
                    <taxRegionNumber></taxRegionNumber>
                    <taxRegionName></taxRegionName>
                    <customerTaxId></customerTaxId>
                    <taxInvoiceNumber></taxInvoiceNumber>
                    <serviceDetail>
                        <bankServiceId></bankServiceId>
                        <serviceDescription></serviceDescription>
                        <originalCharge>
                             <amount></amount>
                             <currencyCode></currencyCode>
                        </originalCharge>
                    </serviceDetail>
                    <taxCalculation>
                        <hostCurCode></hostCurCode>
                        <taxHostConversion>
                            <taxableServiceCharge>
                                <amount></amount>
                                <currencyCode></currencyCode>
                            </taxableServiceCharge>
                            <taxableServiceChargeHost>
                                <amount></amount>
                                <currencyCode></currencyCode>
                            </taxableServiceChargeHost>
                        </taxHostConversion>
                        <totalTaxableSvcChargeHostAmount>
                            <amount></amount>
                            <currencyCode></currencyCode>
                        </totalTaxableSvcChargeHostAmount>
                        <taxCalculationList>
                            <taxIdentificationNumber></
taxIdentificationNumber>
                            <taxIdentifierDescription></
taxIdentifierDescription>
                            <taxIdentifierRate></taxIdentifierRate>
                            <taxIdentifierTotalTaxAmount>
                                <amount></amount>
                                <currencyCode></currencyCode>
                            </taxIdentifierTotalTaxAmount>
                        </taxCalculationList>
                        <totalTaxAmount>
                            <amount></amount>
                            <currencyCode></currencyCode>
                        </totalTaxAmount>
                    </taxCalculation>
                    <settlementAmount>
                        <amount></amount>
                        <currencyCode></currencyCode>
                    </settlementAmount>
                    <taxDueToRegion>
                        <amount></amount>
                        <currencyCode></currencyCode>
                    </taxDueToRegion>
                </taxRegion>
```

Copyright © 2024, Oracle and/or its affiliates.

```
            </taxDetails>
            <serviceAdjustment>
                <Type></Type>
                <description></description>
                <serviceAdjustmentAmt>
                    <amount></amount>
                    <currencyCode></currencyCode>
                </serviceAdjustmentAmt>
                <adjustmentErrorDate></adjustmentErrorDate>
                <serviceAdjustmentID></serviceAdjustmentID>
                <newCharge>
                    <amount></amount>
                    <currencyCode></currencyCode>
                </newCharge>
            </serviceAdjustment>
        </statement>
    </electronicStatement>
</Twist>
```

Related Topics

For more information on...	See...
Tags Used in the TWIST Format	Tags Used in the TWIST Format on page 622
Field Mapping for the TWIST Format	Field Mapping for the TWIST Format on page 602
Sample Statement in the TWIST Format	Sample Statement in the TWIST Format on page 637

Field Mapping for the CAMT Format

Note: We recommend you to refer the CAMT format in parallel while understanding the below mentioned tags. This will help you to understand how the tags are nested in the CAMT format.

The following table shows how each tag is mapped to a field for extracting data in the camt.086 format:

Tag Name	Table	Field	Comments
Document	-	-	-
BkSvcsBllgStmt	-	-	-
RptHdr	-	-	-
RptId	-	-	-
MsgPgntn	-	-	-
PgNb	-	-	-
LastPgInd	-	-	-
BllgStmtGrp	-	-	-
GrpId	-	-	-
Sndr	-	-	The sender person is derived from the **SENDPER** characteristic of the division to which the account belongs.

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Table	Field	Comments
Nm	CI_PER_NAME	ENTITY_NAME	The data is derived from the sender's record where NAME_TYPE_FLG='PRIM'.
LglNm	CI_PER_NAME	ENTITY_NAME	The data is derived from the sender's record where NAME_TYPE_FLG='LGAL'.
PstlAdr	-	-	-
AdrTp	**Note:** The field mapping is not done for these tags. The implementation team will have to map these tags to the appropriate fields.		
Dept			
SubDept			
StrtNm			
BldgNb			
PstCd	CI_PER	POSTAL	-
TwnNm	CI_PER	CITY	-
CtrySubDvsn	CI_PER	STATE	-
Ctry	CI_PER	COUNTRY	-
AdrLine	CI_PER	ADDRESS 1 to ADDRESS 4	-
Flr	**Note:** The field mapping is not done for these tags. The implementation team will have to map these tags to the appropriate fields.		
PstBx			
BldgNm			
Room			
Id	-	-	-
FIId	-	-	-
BICFI	CI_PER_ID	PER_ID_NBR	The data is derived from the sender's record where ID_TYPE_CD='BIC'.
CtryOfRes	CI_PER	COUNTRY	-
CtctDtls	-	-	-
NmPrfx	-	-	The data is derived from the **NAMEPRFX** characteristic of the sender's contact person.
Nm	CI_PER_NAME	ENTITY_NAME	The data is derived from the sender's contact person's record where NAME_TYPE_FLG='PRIM'.
PhneNb	CI_PER_PHONE	COUNTRY_CODE|| PHONE	The data is derived from the sender's contact

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Table	Field	Comments
			person's record where PHONE_TYPE_CD='BUS'.
MobNb	CI_PER_PHONE	COUNTRY_CODE|| PHONE	The data is derived from the sender's contact person's record where PHONE_TYPE_CD='MOB'.
FaxNb	CI_PER_PHONE	COUNTRY_CODE|| PHONE	The data is derived from the sender's contact person's record where PHONE_TYPE_CD='FAX'.
EmailAdr	CI_PER	EMAILID	-
SndrIndvCtct	-	-	-
NmPrfx	-	-	The data is derived from the **NAMEPRFX** characteristic of the sender's individual contact person.
Nm	CI_PER_NAME	ENTITY_NAME	The data is derived from the sender's individual contact person's record where NAME_TYPE_FLG='PRIM'.
PhneNb	CI_PER_PHONE	COUNTRY_CODE|| PHONE	The data is derived from the sender's individual contact person's record where PHONE_TYPE_CD='BUS'.
MobNb	CI_PER_PHONE	COUNTRY_CODE|| PHONE	The data is derived from the sender's individual contact person's record where PHONE_TYPE_CD='MOB'.
FaxNb	CI_PER_PHONE	COUNTRY_CODE|| PHONE	The data is derived from the sender's individual contact person's record where PHONE_TYPE_CD='FAX'.
EmailAdr	CI_PER	EMAILID	-
Rcvr	-	-	The receiver is derived from the account for which the bill is created. The system derives the person who is the main customer of the account.
Nm	CI_PER_NAME	ENTITY_NAME	The data is derived from the receiver's record where NAME_TYPE_FLG='PRIM'.
LglNm	CI_PER_NAME	ENTITY_NAME	The data is derived from the receiver's record where NAME_TYPE_FLG='LGAL'.
PstlAdr	-	-	-

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Table	Field	Comments
AdrTp	-	-	If the bill address source is set to **PER**, then the address type is set to **Billing**. However, if the bill address source is set to **ACOV**, then the address type is set to **Override**.
Dept	**Note:** The field mapping is not done for these tags. The implementation team will have to map these tags to the appropriate fields.		
SubDept			
StrtNm			
BldgNb			
PstCd	CI_PER	POSTAL	-
TwnNm	CI_PER	CITY	-
CtrySubDvsn	CI_PER	STATE	-
Ctry	CI_PER	COUNTRY	-
AdrLine	CI_PER	ADDRESS 1 to ADDRESS 4	-
Flr	**Note:** The field mapping is not done for these tags. The implementation team will have to map these tags to the appropriate fields.		
PstBx			
BldgNm			
Room			
Id	-	-	Depending on the value defined in the **ORGORFI** characteristic of the receiver, the data appears within the **OrgId** or **FIId** tag. If the value is set to **ORG**, the organization identification appears within the **OrgId** tag. However, if the value is set to **FI**, the financial institution identification appears within the **FIId** tag.
OrgId	-	-	-
AnyBIC	CI_PER_ID	PER_ID_NBR	The data is derived from the receiver's record where ID_TYPE_CD='BIC'.
FIId	-	-	-
BICFI	CI_PER_ID	PER_ID_NBR	The data is derived from the receiver's record where ID_TYPE_CD='BIC'.
CtryOfRes	CI_PER	COUNTRY	-

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Table	Field	Comments
CtctDtls	-	-	-
NmPrfx	-	-	The data is derived from the **NAMEPRFX** characteristic of the receiver's contact person.
Nm	CI_PER_NAME	ENTITY_NAME	The data is derived from the receiver's contact person's record where NAME_TYPE_FLG='PRIM'.
PhneNb	CI_PER_PHONE	COUNTRY_CODE|| PHONE	The data is derived from the receiver's contact person's record where PHONE_TYPE_CD='BUS'.
MobNb	CI_PER_PHONE	COUNTRY_CODE|| PHONE	The data is derived from the receiver's contact person's record where PHONE_TYPE_CD='MOB'.
FaxNb	CI_PER_PHONE	COUNTRY_CODE|| PHONE	The data is derived from the receiver's contact person's record where PHONE_TYPE_CD='FAX'.
EmailAdr	CI_PER	EMAILID	-
RcvrIndvCtct	-	-	-
NmPrfx	-	-	The data is derived from the **NAMEPRFX** characteristic of the receiver's individual contact person.
Nm	CI_PER_NAME	ENTITY_NAME	The data is derived from the receiver's individual contact person's record where NAME_TYPE_FLG='PRIM'.
PhneNb	CI_PER_PHONE	COUNTRY_CODE|| PHONE	The data is derived from the receiver's individual contact person's record where PHONE_TYPE_CD='BUS'.
MobNb	CI_PER_PHONE	COUNTRY_CODE|| PHONE	The data is derived from the receiver's individual contact person's record where PHONE_TYPE_CD='MOB'.
FaxNb	CI_PER_PHONE	COUNTRY_CODE|| PHONE	The data is derived from the receiver's individual contact person's record where PHONE_TYPE_CD='FAX'.
EmailAdr	CI_PER	EMAILID	-
BllgStmt	-	-	-

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Table	Field	Comments
StmtId	-	-	Stores System Generated Value
FrToDt	-	-	-
FrDt	-	-	Stores Bill Extraction From Date
ToDt	-	-	Stores Bill Extraction To Date
CreDtTm	-	-	Stores Statement Creation Date and Time
Sts	-	-	Stores Statement Status; Valid Values are - **Original** and **Regenerated**
AcctChrtcs	-	-	-
AcctLvl	-	-	-
CshAcct	-	-	-
Id	-	-	-
IBAN	CI_ACCT_NBR	ACCT_NBR	The data is derived from the record where ACCT_NBR_TYPE_CD='IBAN'.
Tp	-	-	-
Cd	CI_ACCT	CUST_CL_CD	-
Prtry	CI_CUST_CL_L	DESCR	-
Ccy	CI_ACCT	CURRENCY_CD	-
AcctSvcr	-	-	The account servicer person is derived from the **SRVPER** characteristic of the division to which the account belongs.
FinInstnId	-	-	-
BICFI	CI_PER_ID	PER_ID_NBR	The data is derived from the account servicer person's record where ID_TYPE_CD='BIC'.
Nm	CI_PER_NAME	ENTITY_NAME	The data is derived from the account servicer person's record where NAME_TYPE_FLG='PRIM'.
PstlAdr	-	-	-
Dept	**Note:** The field mapping is not done for these tags. The implementation team will have to map these tags to the appropriate fields.		
SubDept			
StrtNm			

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Table	Field	Comments
BldgNb			
PstCd	CI_PER	POSTAL	-
TwnNm	CI_PER	CITY	-
CtrySubDvsn	CI_PER	STATE	-
Ctry	CI_PER	COUNTRY	-
AdrLine	CI_PER	ADDRESS 1 to ADDRESS 4	-
CompstnMtd	-	-	The valid values are **INVD**, **DEBTD**, and **DDBT**.
DbtAcct	-	-	-
IBAN	CI_ACCT_APAY	EXT_ACCT_ID	-
DelydDbtDt	CI_BILL	APAY_CRE_DT	-
SttlmAdvc	CI_ACCT_MSG	-	The data is derived from the record where BILL_MSG_TYPE_FLG='S'.
AcctBalCcyCd	CI_ACCT	CURRENCY_CD	-
SttlmCcyCd	CI_ACCT	CURRENCY_CD	-
HstCcyCd	CI_CIS_DIVISION	BASE_CURRENCY_CD	-
Tax	-	-	-
ClctnMtd	-	-	-
Rgn	-	-	The data is derived from the **TAXREG** characteristic of the account.
AcctSvcrCtct	-	-	-
NmPrfx	-	-	The data is derived from the **NAMEPRFX** characteristic of the account servicer's contact person.
Nm	CI_PER_NAME	ENTITY_NAME	The data is derived from the account servicer's contact person's record where NAME_TYPE_FLG='PRIM'.
PhneNb	CI_PER_PHONE	COUNTRY_CODE|| PHONE	The data is derived from the account servicer's contact person's record where PHONE_TYPE_CD='BUS' or 'HOME'.
MobNb	CI_PER_PHONE	COUNTRY_CODE|| PHONE	The data is derived from the account servicer's contact person's record where PHONE_TYPE_CD='BUS' or 'HOME'.

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Table	Field	Comments
FaxNb	CI_PER_PHONE	COUNTRY_CODE|| PHONE	The data is derived from the account servicer's contact person's record where PHONE_TYPE_CD='BUS' or 'HOME'.
EmailAdr	CI_PER	EMAILID	-
CcyXchg	-	-	-
SrcCcy	CI_BSEG_CALC_LN, CI_BSEG_CALC	NVL(CI_BSEG_CALC_LN.PRIC_CCY_CD, CI_BSEG_CALC.CURRENCY_CD)	
TrgtCcy	CI_BSEG_CALC	CURRENCY_CD	-
XchgRate	CI_BSEG_CALC_LN	EXCHRATE	-
UnitCcy	CI_BSEG_CALC	CURRENCY_CD	-
QtnDt	CI_BILL	NVL(BILL.BILL_DT, BILL.CRE_DTTM)	-
Svc	-	-	This section is added multiple times in the statement when different services are billed to the receiver (i.e. customer) in the statement cycle.
SvcDtl	-	-	-
BkSvc	-	-	-
Id	CI_BILL_CHG	PRICEITEM_CD	-
Desc	CI_PRICEITEM_L	DESCR	-
SvcTp	CI_PRICEITEM_REL	MAX(PRICEITEM_PAR_CD)	The data is derived from the record where the price item is associated with the parent price item using the PRICEITEM_REL_TYPE_FLG='PROD'.
Vol	CI_BSEG_CALC_LN	BILL_SQ	-
Pric	-	-	-
Ccy	CI_BSEG_CALC_LN, CI_BSEG_CALC	NVL(CI_BSEG_CALC_LN.PRIC_CCY_CD, CI_BSEG_CALC.CURRENCY_CD)	
UnitPric	-	-	-
Amt	CI_BSEG_CALC_LN	VALUE_AMT, PRICE_AMT	-
Sgn	-	-	The data is derived from VALUE_AMT or PRICE_AMT.
Mtd	-	-	The method is derived. The valid values are - **UPRC**,

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Table	Field	Comments
			FCHG, **MCHG**, **MXRD**, **TIER** and **TPRC**.
PmtMtd	-	-	The payment method is derived. The valid values are **FLAT**, **PVCH**, **INVS**, and **FREE**.
OrgnlChrgSttlmAmt	-	-	-
Amt	CI_TXN_CALC_LN	CALC_AMT	-
Sgn	-	-	The data is derived from CALC_AMT.
TaxRgn	-	-	-
RgnNb	-	-	The data is derived from the **TAXREG** characteristic of the account.
RgnNm	-	-	The data is derived from the **TAXREG** characteristic of the account.
CstmrTaxId	-	-	The data is derived from the **TAXID** characteristic of the account.
InvcNb	CI_BILL	BILL_ID	-
MtdC	-	-	-
SvcDtl	-	-	This section is added multiple times in the statement when different services are billed to the receiver (i.e. customer) in the statement cycle.
BkSvc	-	-	-
Id	CI_BILL_CHG	PRICEITEM_CD	-
Desc	CI_PRICEITEM_L	DESCR	-
SvcChrgAmt	-	-	-
Amt	CI_BSEG_CALC_LN	CALC_AMT	Here, the amount is equal to the sum of service charge lines in the bill excluding the service charge lines with a tax distribution code (i.e. a distribution code where the **TAXRATE** characteristic is defined). The amount is expressed in the invoice currency of the account.
Sgn	-	-	The data is derived based on the calculated amount.

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Table	Field	Comments
TaxClctn	-	-	-
HstCcy	CI_CIS_DIVISION	BASE_CURRENCY_CD	-
TaxblSvcChrgConvs	-	-	This section is added multiple times in the statement when different tax distribution codes are used for the service.
SrcAmt	-	-	-
Amt	CI_BSEG_CALC_LN	BASE_AMT	Here, the amount is equal to the sum of service charge lines with the same tax distribution code (i.e. a distribution code where the **TAXRATE** characteristic is defined). The amount is expressed in the invoice currency of the account. This calculated amount is used as a source for calculating the tax.
Sgn	-	-	The data is derived based on the calculated amount.
HstAmt	-	-	-
Amt	CI_BSEG_CALC_LN	BASE_AMT	Here, the amount is equal to the sum of service charge lines with the same tax distribution code (i.e. a distribution code where the **TAXRATE** characteristic is defined). The amount is expressed in the division's base currency.
Sgn	-	-	The data is derived based on the calculated amount.
TtlTaxblSvcChrgHstAmt	-	-	-
Amt	-	-	Here, the amount is equal to the sum of all taxable service charge lines. The amount is expressed in the division's base currency.
Sgn	-	-	The data is derived based on the calculated amount.
TaxId	-	-	This section is added multiple times in the statement when different tax distribution codes are used for the service.

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Table	Field	Comments
Nb	CI_BSEG_CALC_LN	DST_ID	-
Desc	CI_DST_CODE_L	DESCR	-
Rate	-	-	The data is derived from the **TAXRATE** characteristic of the distribution code.
TtlTaxAmt	-	-	-
Amt	CI_BSEG_CALC_LN	CALC_AMT	Here, the amount is equal to the sum of taxes charged due to the same tax distribution code. The amount is expressed in the division's base currency.
Sgn	-	-	The data is derived based on the calculated amount.
TtlTax	-	-	-
Amt	CI_BSEG_CALC_LN	CALC_AMT	Here, the amount is equal to the sum total of all taxes charged to the receiver (i.e. customer). The amount is expressed in the division's base currency.
Sgn	-	-	The data is derived based on the calculated amount.
SttlmAmt	-	-	-
Amt	CI_BSEG_CALC_LN	CALC_AMT	Here, the amount is equal to the sum total of all service charges and taxes billed to the receiver (i.e. customer). The amount is expressed in the invoice currency of the account.
Sgn	-	-	The data is derived based on the calculated amount.
TaxDueToRgn	-	-	-
Amt	CI_BSEG_CALC_LN	CALC_AMT	Here, the amount is equal to the sum total of all taxes charged to the receiver (i.e. customer). The amount is expressed in the invoice currency of the account.
Sgn	-	-	The data is derived based on the calculated amount.
SvcAdjstmnt	-	-	-
Tp	CI_ADJ_TYPE	ADJ_TYPE_CD	-

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Table	Field	Comments
Desc	CI_ADJ_TYPE_L	DESCR	-
Amt	-	-	-
Amt	CI_FT	CUR_AMT	-
Sgn	-	-	The data is derived from CUR_AMT.
ErrDt	CI_ADJ	CRE_DT	-
AdjstmntId	CI_ADJ	ADJ_ID	-
NewChrgAmt	-	-	-
Amt	CI_FT	CUR_AMT	-
Sgn	-	-	The data is derived from CUR_AMT.

Related Topics

For more information on...	See...
CAMT Format	CAMT Format on page 582
Tags Used in the CAMT Format	Tags Used in the CAMT Format on page 609
Sample Statement in the CAMT Format	Sample Statement in the CAMT Format on page 631

Field Mapping for the TWIST Format

Note: We recommend you to refer the TWIST format in parallel while understanding the below mentioned tags. This will help you to understand how the tags are nested in the TWIST format.

The following table shows how each tag is mapped to a field for extracting data in the TWIST format:

Tag Name	Table	Field	Comments
Twist	-	-	-
header	-	-	-
electronicStatement	-	-	-
statementHeader	-	-	-
stmtSender	-	-	The sender person is derived from the **SENDPER** characteristic of the division to which the account belongs.
name	CI_PER_NAME	ENTITY_NAME	The data is derived from the sender's record where NAME_TYPE_FLG='PRIM'.
legalName	CI_PER_NAME	ENTITY_NAME	The data is derived from the sender's record where NAME_TYPE_FLG='LGAL'.
contactInfo	-	-	-
postAddress	-	-	-

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Table	Field	Comments
addressIdentifier	**Note:** The field mapping is not done for these tags. The implementation team will have to map these tags to the appropriate fields.		
departmentName			
streetName			
buildingNum			
postCode	CI_PER	POSTAL	-
city	CI_PER	CITY	-
countrySubDivision	-	-	-
name	CI_PER	STATE	-
country	CI_PER	COUNTRY	-
addressLine	CI_PER	ADDRESS 1 to ADDRESS 4	-
floor	**Note:** The field mapping is not done for these tags. The implementation team will have to map these tags to the appropriate fields.		
postBox			
buildingName			
room			
orgId	-	-	-
orgIdNum	CI_PER_ID	PER_ID_NBR	The data is derived from the sender's record where ID_TYPE_CD='BIC'.
contactInfo	-	-	-
individualContact	-	-	-
name	CI_PER_NAME	ENTITY_NAME	The data is derived from the sender's contact person's record where NAME_TYPE_FLG='PRIM'.
phone	CI_PER_PHONE	COUNTRY_CODE|| PHONE	The data is derived from the sender's contact person's record where PHONE_TYPE_CD='BUS'.
fax	CI_PER_PHONE	COUNTRY_CODE|| PHONE	The data is derived from the sender's contact person's record where PHONE_TYPE_CD='FAX'.
Email	CI_PER	EMAILID	-
stmtReceiver	-	-	The receiver is derived from the account for which the bill is created. The system derives the person who is the main customer of the account.
name	CI_PER_NAME	ENTITY_NAME	The data is derived from the receiver's record where NAME_TYPE_FLG='PRIM'.
legalName	CI_PER_NAME	ENTITY_NAME	The data is derived from the receiver's record where NAME_TYPE_FLG='LGAL'.

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Table	Field	Comments
contactInfo	-	-	-
postAddress	-	-	-
addressIdentifier	-	-	If the bill address source is set to **PER**, then the address type is set to **Billing**. However, if the bill address source is set to **ACOV**, then the address type is set to **Override**.
departmentName	**Note:** The field mapping is not done for these tags. The implementation team will have to map these tags to the appropriate fields.		
streetName			
buildingNum			
postCode	CI_PER	POSTAL	-
city	CI_PER	CITY	-
countrySubDivision	-	-	-
name	CI_PER	STATE	-
country	CI_PER	COUNTRY	-
addressLine	CI_PER	ADDRESS 1 to ADDRESS 4	-
floor	**Note:** The field mapping is not done for these tags. The implementation team will have to map these tags to the appropriate fields.		
postBox			
buildingName			
room			
orgId	-	-	Depending on the value defined in the **ORGORFI** characteristic of the receiver, the data appears within the **OrgId** tag. If the value is set to **ORG**, the organization identification appears within the **OrgId** tag. However, if the value is set to **FI**, the financial institution identification appears within the **OrgId** tag.
orgIdNum	CI_PER_ID	PER_ID_NBR	The data is derived from the receiver's record where ID_TYPE_CD='BIC'.
contactInfo	-	-	-
individualContact	-	-	-
name	CI_PER_NAME	ENTITY_NAME	The data is derived from the receiver's contact person's record where NAME_TYPE_FLG='PRIM'.

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Table	Field	Comments
phone	CI_PER_PHONE	COUNTRY_CODE|| PHONE	The data is derived from the receiver's contact person's record where PHONE_TYPE_CD='BUS'.
fax	CI_PER_PHONE	COUNTRY_CODE|| PHONE	The data is derived from the receiver's contact person's record where PHONE_TYPE_CD='FAX'.
Email	CI_PER	EMAILID	-
statement	-	-	-
account	-	-	-
statementStartDate	-	-	Stores Bill Extraction From Date
statementEndDate	-	-	Stores Bill Extraction To Date
statementProductionDate	-	-	Stores Statement Creation Date and Time
statementStatus	-	-	Stores Statement Status; Valid Values are - **Original** and **Regenerated**
accountLevel	-	-	-
iban	CI_ACCT_NBR	ACCT_NBR	The data is derived from the record where ACCT_NBR_TYPE_CD='IBAN'.
domicileBankIdentifier	CI_PER_ID	PER_ID_NBR	The data is derived from the account servicer person's record where ID_TYPE_CD='BIC'.
compensationMethod	-	-	The valid values are **INVD**, **DEBTD**, and **DDBT**.
debitAccount	CI_ACCT_APAY	EXT_ACCT_ID	-
delayedDebitDate	CI_BILL	APAY_CRE_DT	-
settlementAdvice	CI_ACCT_MSG	-	The data is derived from the record where BILL_MSG_TYPE_FLG='S'.
accountBalanceCurrencyCode	CI_ACCT	CURRENCY_CD	-
settlementCurrencyCode	CI_ACCT	CURRENCY_CD	-
hostCurrencyCode	CI_CIS_DIVISION	BASE_CURRENCY_CD	-
taxCalculationMethod	-	-	-
taxRegionCode	-	-	The data is derived from the **TAXREG** characteristic of the account.
bankContactName	CI_PER_NAME	ENTITY_NAME	The data is derived from the account servicer's contact person's record where NAME_TYPE_FLG='PRIM'.

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Table	Field	Comments
bankContactPhone	CI_PER_PHONE	COUNTRY_CODE|| PHONE	The data is derived from the account servicer's contact person's record where PHONE_TYPE_CD='BUS' or 'HOME'.
bankContactFax	CI_PER_PHONE	COUNTRY_CODE|| PHONE	The data is derived from the account servicer's contact person's record where PHONE_TYPE_CD='BUS' or 'HOME'.
bankContactEmail	CI_PER	EMAILID	-
currencyTranslation	-	-	-
originalCurrency	CI_BSEG_CALC_LN, CI_BSEG_CALC	NVL(CI_BSEG_CALC_LN.PRIC_CCY_CD, CI_BSEG_CALC.CURRENCY_CD)	-
targetCurrency	CI_BSEG_CALC	CURRENCY_CD	-
translationValue	CI_BSEG_CALC_LN	EXCHRATE	-
basis	CI_BSEG_CALC	CURRENCY_CD	-
service	-	-	-
bankServiceID	CI_BILL_CHG	PRICEITEM_CD	-
serviceDescription	CI_PRICEITEM_L	DESCR	-
serviceType	CI_PRICEITEM_REL	MAX(PRICEITEM_PARTID)	The data is derived from the record where the price item is associated with the parent price item using the PRICEITEM_REL_TYPE_FLG='PROD'.
volume	CI_BSEG_CALC_LN	BILL_SQ	-
pricingCurrencyCode	CI_BSEG_CALC_LN, CI_BSEG_CALC	NVL(CI_BSEG_CALC_LN.PRIC_CCY_CD, CI_BSEG_CALC.CURRENCY_CD)	-
unitPrice	-	-	-
amount	CI_BSEG_CALC_LN	VALUE_AMT, PRICE_AMT	-
currencyCode	-	-	The data is derived from VALUE_AMT or PRICE_AMT.
priceMethod	-	-	The method is derived. The valid values are - **UPRC**, **FCHG**, **MCHG**, **MXRD**, **TIER** and **TPRC**.
paymentMethod	-	-	The payment method is derived. The valid values are **FLAT**, **PVCH**, **INVS**, and **FREE**.
originalChargeSettlement	-	-	-
amount	CI_TXN_CALC_LN	CALC_AMT	-
currencyCode	-	-	The data is derived from CALC_AMT.
taxDetails	-	-	-

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Table	Field	Comments
taxRegion	-	-	-
taxRegionNumber	-	-	The data is derived from the **TAXREG** characteristic of the account.
taxRegionName	-	-	The data is derived from the **TAXREG** characteristic of the account.
customerTaxId	-	-	The data is derived from the **TAXID** characteristic of the account.
taxInvoiceNumber	CI_BILL	BILL_ID	-
serviceDetail	-	-	-
bankServiceId	CI_BILL_CHG	PRICEITEM_CD	-
serviceDescription	CI_PRICEITEM_L	DESCR	-
originalCharge	-	-	-
amount	CI_BSEG_CALC_LN	CALC_AMT	Here, the amount is equal to the sum of service charge lines in the bill excluding the service charge lines with a tax distribution code (i.e. a distribution code where the **TAXRATE** characteristic is defined). The amount is expressed in the invoice currency of the account.
currencyCode	-	-	The data is derived based on the calculated amount.
taxCalculation	-	-	-
hostCurCode	CI_CIS_DIVISION	BASE_CURRENCY_CD	-
taxHostConversion	-	-	This section is added multiple times in the statement when different tax distribution codes are used for the service.
taxableServiceCharge	-	-	-
amount	CI_BSEG_CALC_LN	BASE_AMT	Here, the amount is equal to the sum of service charge lines with the same tax distribution code (i.e. a distribution code where the **TAXRATE** characteristic is defined). The amount is expressed in the invoice currency of the account. This calculated amount is used as a source for calculating the tax.
currencyCode	-	-	The data is derived based on the calculated amount.
taxableServiceChargeHost	-	-	-

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Table	Field	Comments
amount	CI_BSEG_CALC_LN	BASE_AMT	Here, the amount is equal to the sum of service charge lines with the same tax distribution code (i.e. a distribution code where the **TAXRATE** characteristic is defined). The amount is expressed in the division's base currency.
currencyCode	-	-	The data is derived based on the calculated amount.
totalTaxableSvcChargeHostAmount	-	-	-
amount	-	-	Here, the amount is equal to the sum of all taxable service charge lines. The amount is expressed in the division's base currency.
currencyCode	-	-	The data is derived based on the calculated amount.
taxCalculationList	-	-	This section is added multiple times in the statement when different tax distribution codes are used for the service.
taxIdentificationNumber	CI_BSEG_CALC_LN	DST_ID	-
taxIdentifierDescription	CI_DST_CODE_L	DESCR	-
taxIdentifierRate	-	-	The data is derived from the **TAXRATE** characteristic of the distribution code.
taxIdentifierTotalTaxAmount	-	-	-
amount	CI_BSEG_CALC_LN	CALC_AMT	Here, the amount is equal to the sum of taxes charged due to the same tax distribution code. The amount is expressed in the division's base currency.
currencyCode	-	-	The data is derived based on the calculated amount.
totalTaxAmount	-	-	-
amount	CI_BSEG_CALC_LN	CALC_AMT	Here, the amount is equal to the sum total of all taxes charged to the receiver (i.e. customer). The amount is expressed in the division's base currency.
currencyCode	-	-	The data is derived based on the calculated amount.
settlementAmount	-	-	-
amount	CI_BSEG_CALC_LN	CALC_AMT	Here, the amount is equal to the sum total of all service charges and taxes billed to the receiver (i.e. customer).

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Table	Field	Comments
			The amount is expressed in the invoice currency of the account.
currencyCode	-	-	The data is derived based on the calculated amount.
taxDueToRegion	-	-	-
amount	CI_BSEG_CALC_LN	CALC_AMT	Here, the amount is equal to the sum total of all taxes charged to the receiver (i.e. customer). The amount is expressed in the invoice currency of the account.
currencyCode	-	-	The data is derived based on the calculated amount.
serviceAdjustment	-	-	-
Type	CI_ADJ_TYPE	ADJ_TYPE_CD	-
description	CI_ADJ_TYPE_L	DESCR	-
serviceAdjustmentAmt	-	-	-
amount	CI_FT	CUR_AMT	-
currencyCode	-	-	The data is derived from CUR_AMT.
adjustmentErrorDate	CI_ADJ	CRE_DT	-
serviceAdjustmentID	CI_ADJ	ADJ_ID	-
newCharge	-	-	-
amount	CI_FT	CUR_AMT	-
currencyCode	-	-	The data is derived from CUR_AMT.

Related Topics

For more information on...	See...
TWIST Format	TWIST Format on page 587
Tags Used in the TWIST Format	Tags Used in the TWIST Format on page 622
Sample Statement in the TWIST Format	Sample Statement in the TWIST Format on page 637

Tags Used in the CAMT Format

Note: We recommend you to refer the CAMT format in parallel while understanding the below mentioned tags. This will help you to understand how the tags are nested in the CAMT format.

The following table lists the tags used in the CAMT format and describes the information which is shown in each tag:

Tag Name	Tag Description
Document	Includes the tags used in the CAMT format.
BkSvcsBllgStmt	Displays the details of the statement including the header information.

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Tag Description
RptHdr	Displays the header information of the statement.
RptId	Displays the header ID.
MsgPgntn	Displays the pagination information of the statement.
PgNb	Displays the page number of the statement.
LastPgInd	Indicates that the page is the last page in the statement. The valid values are: • Yes • No
BllgStmtGrp	Displays the details of the sender and receiver. The system groups the statements where the sender and receiver are the same.
GrpId	Displays the group ID.
Sndr	Displays information about the sender person (i.e. financial institution). **Note:** The sender person is derived from the **SENDPER** characteristic of the division to which the account belongs.
Nm	This tag appears within the following tags: **If this tag is used within...** — **Then the purpose of the Nm tag is...** Sndr — Displays the primary name of the financial institution. CtctDtls — Displays the primary name of the sender's or receiver's contact person. SndrIndvCtct — Displays the primary name of the sender's individual contact person. Rcvr — Displays the primary name of the financial institution or organization. RcvrIndvCtct — Displays the primary name of the receiver's individual contact person. FinInstnId — Displays the primary name of the account servicer. **Note:** The account servicer is a domicile bank (i.e. a branch) in which the account actually resides. An account servicer is defined as a person in ORMB. The account servicer for an account is derived from the **SRVPER** characteristic of the division to which the account belongs. AcctSvcrCtct — Displays the primary name of the account servicer's contact person.
LglNm	This tag appears within the following tags: **If this tag is used within...** — **Then the purpose of the LglNm tag is...** Sndr — Displays the legal name of the financial institution. Rcvr — Displays the legal name of the financial institution or organization.
PstlAdr	This tag appears within the following tags:

Copyright © 2024, Oracle and/or its affiliates.

<table>
<tr><th>Tag Name</th><th>Tag Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the PstlAdr tag is...</th></tr>
<tr><td>Sndr</td><td>Displays the address of the financial institution.</td></tr>
<tr><td>Rcvr</td><td>Displays the address of the financial institution or organization.</td></tr>
<tr><td>FinInstnId</td><td>Displays the address of the account servicer.</td></tr>
</table>
</td></tr>
<tr><td>AdrTp</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the AdrTp tag is...</th></tr>
<tr><td>PstlAdr (Sender)</td><td>At present, no information appears in this tag. The implementation team can write the custom logic to display the required information in this tag.</td></tr>
<tr><td>PstlAdr (Receiver)</td><td>The information shown in this tag varies depending on the bill address source. If the bill address source is PER, then the tag contains the string ‘Billing’. However, if the bill address source is ACOV, then the tag contains the string ‘Override’.</td></tr>
</table>
</td></tr>
<tr><td>Dept</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the Dept tag is...</th></tr>
<tr><td>PstlAdr</td><td>At present, no information appears in this tag. The implementation team can write the custom logic to display the required information in this tag.</td></tr>
</table>
</td></tr>
<tr><td>SubDept</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the SubDept tag is...</th></tr>
<tr><td>PstlAdr</td><td>At present, no information appears in this tag. The implementation team can write the custom logic to display the required information in this tag.</td></tr>
</table>
</td></tr>
<tr><td>StrtNm</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the StrtNm tag is...</th></tr>
<tr><td>PstlAdr</td><td>At present, no information appears in this tag. The implementation team can write the custom logic to display the required information in this tag.</td></tr>
</table>
</td></tr>
<tr><td>BldgNb</td><td>This tag appears within the following tags:</td></tr>
</table>

Copyright © 2024, Oracle and/or its affiliates.

<table>
<tr><th>Tag Name</th><th>Tag Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the BldgNb tag is...</th></tr>
<tr><td>PstlAdr</td><td>At present, no information appears in this tag. The implementation team can write the custom logic to display the required information in this tag.</td></tr>
</table>
</td></tr>
<tr><td>PstCd</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the PstCd tag is...</th></tr>
<tr><td>PstlAdr</td><td>Displays the zip or postal code of the address.</td></tr>
</table>
</td></tr>
<tr><td>TwnNm</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the TwnNm tag is...</th></tr>
<tr><td>PstlAdr</td><td>Displays the city name.</td></tr>
</table>
</td></tr>
<tr><td>CtrySubDvsn</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the CtrySubDvsn tag is...</th></tr>
<tr><td>PstlAdr</td><td>Displays the state where the city is located.</td></tr>
</table>
</td></tr>
<tr><td>Ctry</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the Ctry tag is...</th></tr>
<tr><td>PstlAdr</td><td>Displays the country where the address is located.</td></tr>
</table>
</td></tr>
<tr><td>AdrLine</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the AdrLine tag is...</th></tr>
<tr><td>PstlAdr</td><td>Displays a concatenated string consisting of address line 1, 2, 3, and 4 separated by a comma.</td></tr>
</table>
</td></tr>
<tr><td>Flr</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the Flr tag is...</th></tr>
<tr><td>PstlAdr</td><td>At present, no information appears in this tag. The implementation team can write the custom logic to display the required information in this tag.</td></tr>
</table>
</td></tr>
<tr><td>PstBx</td><td>This tag appears within the following tags:</td></tr>
</table>

Copyright © 2024, Oracle and/or its affiliates.

<table>
<tr><th>Tag Name</th><th>Tag Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the PstBx tag is...</th></tr>
<tr><td>PstlAdr</td><td>At present, no information appears in this tag. The implementation team can write the custom logic to display the required information in this tag.</td></tr>
</table>
</td></tr>
<tr><td>BldgNm</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the BldgNm tag is...</th></tr>
<tr><td>PstlAdr</td><td>At present, no information appears in this tag. The implementation team can write the custom logic to display the required information in this tag.</td></tr>
</table>
</td></tr>
<tr><td>Room</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the Room tag is...</th></tr>
<tr><td>PstlAdr</td><td>At present, no information appears in this tag. The implementation team can write the custom logic to display the required information in this tag.</td></tr>
</table>
</td></tr>
<tr><td>Id</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the Id tag is...</th></tr>
<tr><td>Sndr</td><td>Displays the identification details of the financial institution.</td></tr>
<tr><td>Rcvr</td><td>Displays the identification details of the financial institution or organization.</td></tr>
<tr><td>CshAcct</td><td>Displays the identification details of the account for which the bill is created.</td></tr>
<tr><td>BkSvc</td><td>Displays the identification details of the service offered by the financial institution. In ORMB terms, it indicates the price item offered to the customer.</td></tr>
</table>
</td></tr>
<tr><td>FIId</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the FIId tag is...</th></tr>
<tr><td>Id</td><td>Displays the identification details of the financial institution.</td></tr>
</table>
</td></tr>
<tr><td>BICFI</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the BICFI tag is...</th></tr>
<tr><td>FIId</td><td>Displays the bank identifier code of the financial institution.</td></tr>
</table>
</td></tr>
</table>

Copyright © 2024, Oracle and/or its affiliates.

<table>
<tr><th>Tag Name</th><th>Tag Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the BICFI tag is...</th></tr>
<tr><td>FinInstnId</td><td>Displays the bank identifier code of the account servicer.</td></tr>
</table>
</td></tr>
<tr><td>CtryOfRes</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the CtryOfRes tag is...</th></tr>
<tr><td>Sndr</td><td>Displays the country from where the affairs of the financial institution are directed.</td></tr>
<tr><td>Rcvr</td><td>Displays the country from where the affairs of the financial institution or organization are directed.</td></tr>
</table>
</td></tr>
<tr><td>CtctDtls</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the CtctDtls tag is...</th></tr>
<tr><td>Sndr</td><td>Displays the contact details of the financial institution.</td></tr>
<tr><td>Rcvr</td><td>Displays the contact details of the financial institution or organization.</td></tr>
</table>
</td></tr>
<tr><td>NmPrfx</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the NmPrfx tag is...</th></tr>
<tr><td>CtctDtls</td><td>Displays the prefix for the sender's or receiver's contact person's name.</td></tr>
<tr><td>SndrIndvCtct</td><td>Displays the prefix for the sender's individual contact person's name.</td></tr>
<tr><td>RcvrIndvCtct</td><td>Displays the prefix for the receiver's individual contact person's name.</td></tr>
<tr><td>AcctSvcrCtct</td><td>Displays the prefix for the account servicer's contact person's name.</td></tr>
</table>
Note: The data is derived from the NAMEPRFX characteristic of the respective person.
</td></tr>
<tr><td>PhneNb</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the PhneNb tag is...</th></tr>
<tr><td>CtctDtls</td><td>Displays the business phone number of the sender's or receiver's contact person.</td></tr>
<tr><td>SndrIndvCtct</td><td>Displays the business phone number of the sender's individual contact person.</td></tr>
<tr><td>RcvrIndvCtct</td><td>Displays the business phone number of the receiver's individual contact person.</td></tr>
<tr><td>AcctSvcrCtct</td><td>Displays the business or home phone number of the account servicer's contact person.</td></tr>
</table>
</td></tr>
</table>

Copyright © 2024, Oracle and/or its affiliates.

<table>
<tr><th>Tag Name</th><th>Tag Description</th></tr>
<tr><td>MobNb</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the MobNb tag is...</th></tr>
<tr><td>CtctDtls</td><td>Displays the mobile phone number of the sender's or receiver's contact person.</td></tr>
<tr><td>SndrIndvCtct</td><td>Displays the mobile phone number of the sender's individual contact person.</td></tr>
<tr><td>RcvrIndvCtct</td><td>Displays the mobile phone number of the receiver's individual contact person.</td></tr>
<tr><td>AcctSvcrCtct</td><td>Displays the business or home phone number of the account servicer's contact person.</td></tr>
</table></td></tr>
<tr><td>FaxNb</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the FaxNb tag is...</th></tr>
<tr><td>CtctDtls</td><td>Displays the fax number of the sender's or receiver's contact person.</td></tr>
<tr><td>SndrIndvCtct</td><td>Displays the fax number of the sender's individual contact person.</td></tr>
<tr><td>RcvrIndvCtct</td><td>Displays the fax number of the receiver's individual contact person.</td></tr>
<tr><td>AcctSvcrCtct</td><td>Displays the business or home phone number of the account servicer's contact person.</td></tr>
</table></td></tr>
<tr><td>EmailAdr</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the EmailAdr tag is...</th></tr>
<tr><td>CtctDtls</td><td>Displays the e-mail address of the sender's or receiver's contact person.</td></tr>
<tr><td>SndrIndvCtct</td><td>Displays the e-mail address of the sender's individual contact person.</td></tr>
<tr><td>RcvrIndvCtct</td><td>Displays the e-mail address of the receiver's individual contact person.</td></tr>
<tr><td>AcctSvcrCtct</td><td>Displays the e-mail address of the account servicer's contact person.</td></tr>
</table></td></tr>
<tr><td>SndrIndvCtct</td><td>Displays details of the sender's individual contact person.</td></tr>
<tr><td>Rcvr</td><td>Displays information about the receiver person (i.e. financial institution or organization). The receiver is derived from the account for which the bill is created. The system derives the person who is the main customer of the account.</td></tr>
<tr><td>OrgId</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the OrgId tag is...</th></tr>
<tr><td>Id</td><td>Displays the identification details of the organization.</td></tr>
</table></td></tr>
<tr><td>AnyBIC</td><td>This tag appears within the following tags:</td></tr>
</table>

Copyright © 2024, Oracle and/or its affiliates.

<table>
<tr><th>Tag Name</th><th>Tag Description</th></tr>
<tr><td></td><td><table><tr><th>If this tag is used within...</th><th>Then the purpose of the AnyBIC tag is...</th></tr><tr><td>OrgId</td><td>Displays the bank identifier code of the organization.</td></tr></table></td></tr>
<tr><td>RcvrIndvCtct</td><td>Displays the details of the receiver's individual contact person.</td></tr>
<tr><td>BllgStmt</td><td>Displays the details of the statement. A statement includes all service charges that occurred during a statement cycle.</td></tr>
<tr><td>StmtId</td><td>Displays the statement ID.</td></tr>
<tr><td>FrToDt</td><td>Displays the date range for which the statement is issued.</td></tr>
<tr><td>FrDt</td><td>This tag appears within the following tags:<table><tr><th>If this tag is used within...</th><th>Then the purpose of the FrDt tag is...</th></tr><tr><td>FrToDt</td><td>Displays the date from when the bills are considered for extraction in the statement.</td></tr></table>Note: The system sets the from date to the last processing date of the statement cycle plus one day.</td></tr>
<tr><td>ToDt</td><td>This tag appears within the following tags:<table><tr><th>If this tag is used within...</th><th>Then the purpose of the ToDt tag is...</th></tr><tr><td>FrToDt</td><td>Displays the date till when the bills are considered for extraction in the statement.</td></tr></table>Note: The system sets the to date to the statement cycle's processing date which is earlier than the batch business date and is in the Pending status.</td></tr>
<tr><td>CreDtTm</td><td>Displays the date and time when the statement is created.</td></tr>
<tr><td>Sts</td><td>Displays the status of the statement. The valid values are:
Original
Regenerated</td></tr>
<tr><td>AcctChrtcs</td><td>Displays the details of the account for which the bills are extracted.</td></tr>
<tr><td>AcctLvl</td><td>Contains a hard-coded string ‘DETL’.</td></tr>
<tr><td>CshAcct</td><td>Displays the basic details of the account.</td></tr>
<tr><td>IBAN</td><td>This tag appears within the following tags:<table><tr><th>If this tag is used within...</th><th>Then the purpose of the IBAN tag is...</th></tr><tr><td>Id</td><td>Displays the international bank account number.</td></tr><tr><td>DbtAcct</td><td>Displays the international bank account number.</td></tr></table></td></tr>
<tr><td>Tp</td><td>This tag appears within the following tags:</td></tr>
</table>

Copyright © 2024, Oracle and/or its affiliates.

<table>
<tr><th>Tag Name</th><th>Tag Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the Tp tag is...</th></tr>
<tr><td>CshAcct</td><td>Displays the details of the customer class to which the account belongs.</td></tr>
<tr><td>SvcAdjstmnt</td><td>Displays the adjustment type using which the adjustment is created.</td></tr>
</table>
</td></tr>
<tr><td>Cd</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the Cd tag is...</th></tr>
<tr><td>Tp</td><td>Displays the customer class to which the account belongs.</td></tr>
</table>
</td></tr>
<tr><td>Prtry</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the Prtry tag is...</th></tr>
<tr><td>Tp</td><td>Displays the description of the customer class.</td></tr>
</table>
</td></tr>
<tr><td>Ccy</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the Ccy tag is...</th></tr>
<tr><td>CshAcct</td><td>Displays the invoice currency of the account.</td></tr>
<tr><td>Pric</td><td>Displays the currency in which the pricing is defined.</td></tr>
</table>
</td></tr>
<tr><td>AcctSvcr</td><td>Displays the details of the account servicer.

Note: The account servicer is a domicile bank (i.e. a branch) in which the account actually resides. An account servicer is defined as a person in ORMB. The account servicer for an account is derived from the SRVPER characteristic of the division to which the account belongs.</td></tr>
<tr><td>FinInstnId</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the FinInstnId tag is...</th></tr>
<tr><td>AcctSvcr</td><td>Displays the identification details of the account servicer.</td></tr>
</table>
</td></tr>
<tr><td>CompstnMtd</td><td>If the automatic payment feature is not configured for the account, the compensation method is set to INVD (i.e. Invoiced). However, if the automatic payment feature is configured for the account and the automatic payment creation date is earlier than the batch business date, the compensation method is set to DEBTD (i.e. Debited). And, if the automatic payment feature is configured for the account and the automatic payment creation date is later than the batch business date, the compensation method is set to DDBT (i.e. Delayed Debit).</td></tr>
<tr><td>DbtAcct</td><td>Displays the details of the account from where the charges and taxes should be automatically debited. Usually, this account is different from the account which has availed the services.</td></tr>
<tr><td>DelydDbtDt</td><td>Displays the date when the charges and taxes are automatically debited.</td></tr>
</table>

Copyright © 2024, Oracle and/or its affiliates.

<table>
<tr><th>Tag Name</th><th>Tag Description</th></tr>
<tr><td>SttlmAdvc</td><td>Displays the message advising the customer about the settlement of charges and taxes which are due.

Note: Here, a bill message whose bill message type is set to S appears in this tag.</td></tr>
<tr><td>AcctBalCcyCd</td><td>Displays the currency in which the account balance is shown.</td></tr>
<tr><td>SttlmCcyCd</td><td>Displays the currency in which the account is settled.</td></tr>
<tr><td>HstCcyCd</td><td>Displays the division's base currency which is used during currency conversion.</td></tr>
<tr><td>Tax</td><td>Displays the tax information of the account.</td></tr>
<tr><td>ClctnMtd</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the ClctnMtd tag is...</th></tr>
<tr><td>Tax</td><td>Displays the calculation method which indicates how the taxes are applied on the account. The valid value is Method C.</td></tr>
</table></td></tr>
<tr><td>Rgn</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the Rgn tag is...</th></tr>
<tr><td>Tax</td><td>Indicates the tax region in which the account resides.</td></tr>
</table>
Note: The tax region is derived from the TAXREG characteristic of the account.</td></tr>
<tr><td>AcctSvcrCtct</td><td>Displays the details of the account servicer's contact person. At present, here, the system displays the details of the account servicer.

Note: The account servicer's contact person is an individual whom you can contact at the financial institution's location regarding problems of a business nature.</td></tr>
<tr><td>CcyXchg</td><td>Displays the details of an exchange rate which is used in the bill.</td></tr>
<tr><td>SrcCcy</td><td>Displays the currency from which the amount is converted.</td></tr>
<tr><td>TrgtCcy</td><td>Displays the currency to which the amount is converted.</td></tr>
<tr><td>XchgRate</td><td>Displays the exchange rate.</td></tr>
<tr><td>UnitCcy</td><td>Displays the currency in which the exchange rate is defined.</td></tr>
<tr><td>QtnDt</td><td>Displays the date when the exchange rate is used.</td></tr>
<tr><td>Svc</td><td>Displays the details of the service (i.e. price item) offered by the financial institution.</td></tr>
<tr><td>SvcDtl</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the SvcDtl tag is...</th></tr>
<tr><td>Svc</td><td>Displays the basic details of the price item.</td></tr>
<tr><td>MtdC</td><td>Displays the basic details of the price item.</td></tr>
</table></td></tr>
</table>

Copyright © 2024, Oracle and/or its affiliates.

<table>
<tr><th>Tag Name</th><th>Tag Description</th></tr>
<tr><td></td><td>Note: This tag is added multiple times in the statement when different services are billed to the receiver (i.e. customer) in the statement cycle.</td></tr>
<tr><td>BkSvc</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the BkSvc tag is...</th></tr>
<tr><td>SvcDtl</td><td>Displays the identification details of the price item.</td></tr>
</table></td></tr>
<tr><td>Desc</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the Desc tag is...</th></tr>
<tr><td>BkSvc</td><td>Displays the description of the price item.</td></tr>
<tr><td>TaxId</td><td>Displays the description of the distribution code.</td></tr>
<tr><td>SvcAdjstmnt</td><td>Displays the description of the adjustment type.</td></tr>
</table></td></tr>
<tr><td>SvcTp</td><td>Displays the parent price item to which the price item is associated.</td></tr>
<tr><td>Vol</td><td>Displays the quantity of the service (i.e. price item) which is charged in the bill.</td></tr>
<tr><td>Pric</td><td>Displays the details of the pricing which is used to the charge the customer.</td></tr>
<tr><td>UnitPric</td><td>Displays the price item pricing used per unit to calculate the charge in the pricing currency.</td></tr>
<tr><td>Amt</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the Amt tag is...</th></tr>
<tr><td>UnitPric</td><td>Displays the rate per unit in the pricing currency.</td></tr>
<tr><td>OrgnlChrgSttlmAmt</td><td>Displays the calculated charge in the invoice currency.</td></tr>
<tr><td>SvcChrgAmt</td><td>Displays the sum of service charge lines in the bill excluding the service charge lines with a tax distribution code (i.e. a distribution code where the TAXRATE characteristic is defined). The amount is expressed in the invoice currency of the account.</td></tr>
<tr><td>SrcAmt</td><td>Displays the sum of service charge lines with the same tax distribution code (i.e. a distribution code where the TAXRATE characteristic is defined). The amount is expressed in the invoice currency of the account. This calculated amount is used as a source for calculating the tax.</td></tr>
<tr><td>HstAmt</td><td>Displays the sum of service charge lines with the same tax distribution code (i.e. a distribution code where the TAXRATE characteristic is defined). The amount is expressed in the division's base currency.</td></tr>
<tr><td>TtlTaxblSvcChrgHstAmt</td><td>Displays the sum of all taxable service charge lines. The amount is expressed in the division's base currency.</td></tr>
<tr><td>TtlTaxAmt</td><td>Displays the sum of taxes charged due to the same tax distribution code. The amount is expressed in the division's base currency.</td></tr>
</table></td></tr>
</table>

Copyright © 2024, Oracle and/or its affiliates.

<table>
<tr><th>Tag Name</th><th>Tag Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the Amt tag is...</th></tr>
<tr><td>TtlTax</td><td>Displays the sum total of all taxes charged to the receiver (i.e. customer). The amount is expressed in the division's base currency.</td></tr>
<tr><td>SttlmAmt</td><td>Displays the sum total of all service charges and taxes billed to the receiver (i.e. customer). The amount is expressed in the invoice currency of the account.</td></tr>
<tr><td>TaxDueToRgn</td><td>Displays the sum total of all taxes charged to the receiver (i.e. customer). The amount is expressed in the invoice currency of the account.</td></tr>
<tr><td>Amt</td><td>Displays the adjustment amount in the invoice currency.</td></tr>
<tr><td>NewChrgAmt</td><td>Displays the adjustment amount billed to the customer.</td></tr>
</table>
</td></tr>
<tr><td>Sgn</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the Sgn tag is...</th></tr>
<tr><td>UnitPric</td><td>Indicates whether the rate is positive or negative.</td></tr>
<tr><td>OrgnlChrgSttlmAmt</td><td>Indicates whether the calculated charge is positive or negative.</td></tr>
<tr><td>SvcChrgAmt</td><td>Indicates whether the calculated amount is positive or negative.</td></tr>
<tr><td>SrcAmt</td><td>Indicates whether the calculated amount is positive or negative.</td></tr>
<tr><td>HstAmt</td><td>Indicates whether the calculated amount is positive or negative.</td></tr>
<tr><td>TtlTaxblSvcChrgHstAmt</td><td>Indicates whether the calculated amount is positive or negative.</td></tr>
<tr><td>TtlTaxAmt</td><td>Indicates whether the calculated amount is positive or negative.</td></tr>
<tr><td>TtlTax</td><td>Indicates whether the calculated amount is positive or negative.</td></tr>
<tr><td>SttlmAmt</td><td>Indicates whether the calculated amount is positive or negative.</td></tr>
<tr><td>TaxDueToRgn</td><td>Indicates whether the calculated amount is positive or negative.</td></tr>
<tr><td>Amt</td><td>Indicates whether the adjustment amount is positive or negative.</td></tr>
<tr><td>NewChrgAmt</td><td>Indicates whether the adjustment amount is positive or negative.</td></tr>
</table>
The valid values are:

True - Indicates that the amount is positive.
False - Indicates that the amount is negative.

</td></tr>
<tr><td>Mtd</td><td>Indicates the type of price. The valid values are:

UPRC - Indicates that the price is a unit price.
FCHG - Indicates that the price is a flat price.
MCHG - Indicates that the price is a minimum price.
MXRD - Indicates that the price is a maximum price.
TIER - Indicates that the price is a step tiered price.
TPRC - Indicates that the price is a threshold tiered price.

</td></tr>
</table>

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Tag Description
PmtMtd	Displays the type of charge created for the service. The valid values are: • **FLAT** - Indicates that the charge is an SQI-based billable charge. • **PVCH** - Indicates that the charge is a pass through memo charge. • **INVS** - Indicates that the charge is a pass through billable charge. • **FREE** - Indicates that the charge is a 100% discounted charge.
OrgnlChrgSttlmAmt	Displays the details of the calculated charge.
TaxRgn	Displays the details of the tax region.
RgnNb	Displays the tax region in which the account resides. **Note:** The tax region is derived from the **TAXREG** characteristic of the account.
RgnNm	Displays the tax region in which the account resides. **Note:** The tax region is derived from the **TAXREG** characteristic of the account.
CstmrTaxId	Displays the tax identification number of the receiver (i.e. financial institution or organization). **Note:** The tax ID is derived from the **TAXID** characteristic of the account.
InvcNb	Displays the bill ID.
MtdC	Indicates how the taxes charged on the services are calculated.
SvcChrgAmt	Displays the sum of service charge lines in the bill excluding the service charge lines with a tax distribution code.
TaxClctn	Displays the details of tax calculation.
HstCcy	Displays the division's base currency which is used as the host currency.
TaxblSvcChrgConvs	Displays the sum of service charge lines with a tax distribution code. This tag is added multiple times in the statement when different tax distribution codes are used for the service.
SrcAmt	Displays the sum of service charge lines with the same tax distribution code. The amount is expressed in the invoice currency.
HstAmt	Displays the sum of service charge lines with the same tax distribution code. The amount is expressed in the division's base currency.
TtlTaxblSvcChrgHstAmt	Displays the sum of all taxable service charge lines in the division's base currency.
TaxId	Displays the details of the tax applied on the service.
Nb	Displays the distribution code where tax is applicable. In ORMB terms, it is a distribution code where the **TAXRATE** characteristic is defined.
Rate	Displays the rate at which the tax is calculated. **Note:** The tax rate is derived from the **TAXRATE** characteristic of the distribution code.
TtlTaxAmt	Displays the sum of taxes charged due to the tax distribution code.
TtlTax	Displays the sum total of all taxes charged to the receiver (i.e. customer).
SttlmAmt	Displays the sum total of all service charges and taxes billed to the receiver (i.e. customer).
TaxDueToRgn	Displays the sum total of all taxes charged to the receiver (i.e. customer).

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Tag Description
SvcAdjstmnt	Displays the details of the adjustment which is created for the service and posted onto the bill.
Amt	Displays the adjustment amount.
ErrDt	Displays the date when the adjustment is created.
AdjstmntId	Displays the adjustment ID.
NewChrgAmt	Displays the adjustment amount billed to the customer.

Related Topics

For more information on...	See...
CAMT Format	CAMT Format on page 582
Field Mapping for the CAMT Format	Field Mapping for the CAMT Format on page 591
Sample Statement in the CAMT Format	Sample Statement in the CAMT Format on page 631

Tags Used in the TWIST Format

Note: We recommend you to refer the TWIST format in parallel while understanding the below mentioned tags. This will help you to understand how the tags are nested in the TWIST format.

The following table lists the tags used in the TWIST format and describes the information which is shown in each tag:

<table>
<tr><th>Tag Name</th><th>Tag Description</th></tr>
<tr><td>Twist</td><td>Includes the tags used in the TWIST format.</td></tr>
<tr><td>header</td><td>Displays the header ID.</td></tr>
<tr><td>electronicStatement</td><td>Displays the details of the statement including the header information.</td></tr>
<tr><td>stmtSender</td><td>Displays information about the sender person (i.e. financial institution).

Note: The sender person is derived from the SENDPER characteristic of the division to which the account belongs.</td></tr>
<tr><td>name</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the name tag is...</th></tr>
<tr><td>stmtSender</td><td>Displays the primary name of the financial institution.</td></tr>
<tr><td>countrySubDivision</td><td>Indicates the state where the city is located.</td></tr>
<tr><td>individualContact</td><td>Displays the primary name of the sender's or receiver's individual contact person.</td></tr>
<tr><td>stmtReceiver</td><td>Displays the primary name of the financial institution or organization.</td></tr>
</table></td></tr>
<tr><td>legalName</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the legalName tag is...</th></tr>
<tr><td>stmtSender</td><td>Displays the legal name of the financial institution.</td></tr>
</table></td></tr>
</table>

Copyright © 2024, Oracle and/or its affiliates.

<table>
<tr><th>Tag Name</th><th>Tag Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the legalName tag is...</th></tr>
<tr><td>stmtReceiver</td><td>Displays the legal name of the financial institution or organization.</td></tr>
</table>
</td></tr>
<tr><td>contactInfo</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the contactInfo tag is...</th></tr>
<tr><td>stmtSender</td><td>Displays the contact details of the financial institution.</td></tr>
<tr><td>stmtReceiver</td><td>Displays the contact details of the financial institution or organization.</td></tr>
</table>
</td></tr>
<tr><td>postAddress</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the postAddress tag is...</th></tr>
<tr><td>contactInfo</td><td>Displays the address details of the sender or receiver.</td></tr>
</table>
</td></tr>
<tr><td>addressIdentifier</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the addressIdentifier tag is...</th></tr>
<tr><td>postAddress (Sender)</td><td>At present, no information appears in this tag. The implementation team can write the custom logic to display the required information in this tag.</td></tr>
<tr><td>postAddress (Receiver)</td><td>The information shown in this tag varies depending on the bill address source. If the bill address source is PER, then the tag contains the string ‘Billing’. However, if the bill address source is ACOV, then the tag contains the string ‘Override’.</td></tr>
</table>
</td></tr>
<tr><td>departmentName</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the departmentName tag is...</th></tr>
<tr><td>postAddress</td><td>At present, no information appears in this tag. The implementation team can write the custom logic to display the required information in this tag.</td></tr>
</table>
</td></tr>
<tr><td>streetName</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the streetName tag is...</th></tr>
<tr><td>postAddress</td><td>At present, no information appears in this tag. The implementation team can write the custom logic to display the required information in this tag.</td></tr>
</table>
</td></tr>
<tr><td>buildingNum</td><td>This tag appears within the following tags:</td></tr>
</table>

Copyright © 2024, Oracle and/or its affiliates.

<table>
<tr><th>Tag Name</th><th>Tag Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the buildingNum tag is...</th></tr>
<tr><td>postAddress</td><td>At present, no information appears in this tag. The implementation team can write the custom logic to display the required information in this tag.</td></tr>
</table>
</td></tr>
<tr><td>postCode</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the postCode tag is...</th></tr>
<tr><td>postAddress</td><td>Displays the zip or postal code of the address.</td></tr>
</table>
</td></tr>
<tr><td>city</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the city tag is...</th></tr>
<tr><td>postAddress</td><td>Displays the city name.</td></tr>
</table>
</td></tr>
<tr><td>countrySubDivision</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the countrySubDivision tag is...</th></tr>
<tr><td>postAddress</td><td>Displays the state where the city is located.</td></tr>
</table>
</td></tr>
<tr><td>country</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the country tag is...</th></tr>
<tr><td>postAddress</td><td>Displays the country where the address is located.</td></tr>
</table>
</td></tr>
<tr><td>addressLine</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the addressLine tag is...</th></tr>
<tr><td>postAddress</td><td>Displays a concatenated string consisting of address line 1, 2, 3, and 4 separated by a comma.</td></tr>
</table>
</td></tr>
<tr><td>floor</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the floor tag is...</th></tr>
<tr><td>postAddress</td><td>At present, no information appears in this tag. The implementation team can write the custom logic to display the required information in this tag.</td></tr>
</table>
</td></tr>
<tr><td>postBox</td><td>This tag appears within the following tags:</td></tr>
</table>

Copyright © 2024, Oracle and/or its affiliates.

<table>
<tr><th>Tag Name</th><th>Tag Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the postBox tag is...</th></tr>
<tr><td>postAddress</td><td>At present, no information appears in this tag. The implementation team can write the custom logic to display the required information in this tag.</td></tr>
</table>
</td></tr>
<tr><td>buildingName</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the buildingName tag is...</th></tr>
<tr><td>postAddress</td><td>At present, no information appears in this tag. The implementation team can write the custom logic to display the required information in this tag.</td></tr>
</table>
</td></tr>
<tr><td>room</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the room tag is...</th></tr>
<tr><td>postAddress</td><td>At present, no information appears in this tag. The implementation team can write the custom logic to display the required information in this tag.</td></tr>
</table>
</td></tr>
<tr><td>orgId</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the orgId tag is...</th></tr>
<tr><td>stmtSender</td><td>Displays the identification details of the financial institution.</td></tr>
<tr><td>stmtReceiver</td><td>Displays the identification details of the financial institution or organization.</td></tr>
</table>
</td></tr>
<tr><td>orgIdNum</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the orgIdNum tag is...</th></tr>
<tr><td>orgId (Sender)</td><td>Displays the bank identifier code of the financial institution.</td></tr>
<tr><td>orgId (Receiver)</td><td>Displays the bank identifier code of the financial institution or organization.</td></tr>
</table>
</td></tr>
<tr><td>individualContact</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the individualContact tag is...</th></tr>
<tr><td>stmtSender</td><td>Displays the name of the sender's individual contact person.</td></tr>
<tr><td>stmtReceiver</td><td>Displays the name of the receiver's individual contact person.</td></tr>
</table>
</td></tr>
<tr><td>phone</td><td>This tag appears within the following tags:</td></tr>
</table>

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Tag Description
	If this tag is used within... / **Then the purpose of the phone tag is...** contactInfo / Displays the business phone number of the sender's or receiver's individual contact person.
fax	This tag appears within the following tags: **If this tag is used within...** / **Then the purpose of the fax tag is...** contactInfo / Displays the fax number of the sender's or receiver's individual contact person.
Email	This tag appears within the following tags: **If this tag is used within...** / **Then the purpose of the Email tag is...** contactInfo / Displays the e-mail address of the sender's or receiver's individual contact person.
stmtReceiver	Displays information about the receiver person (i.e. financial institution or organization). The receiver is derived from the account for which the bill is created. The system derives the person who is the main customer of the account.
statement	Displays the details of the statement. A statement includes all service charges that occurred during a statement cycle.
account	Displays the details of the account for which the bills are extracted.
statementStartDate	Displays the date from when the bills are considered for extraction in the statement. **Note:** The system sets the from date to the last processing date of the statement cycle plus one day.
statementEndDate	Displays the date till when the bills are considered for extraction in the statement. **Note:** The system sets the to date to the statement cycle's processing date which is earlier than the batch business date and is in the **Pending** status.
statementProductionDate	Displays the date and time when the statement is created.
statementStatus	Displays the status of the statement. The valid values are: • Original • Regenerated
accountLevel	Contains a hard-coded string ‘DETL’.
iban	Displays the international bank account number.
domicileBankIdentifier	Displays the bank identifier code of the account servicer. **Note:** The account servicer is a domicile bank (i.e. a branch) in which the account actually resides. An account servicer is defined as a person in ORMB. The account servicer for an account is derived from the **SRVPER** characteristic of the division to which the account belongs.

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Tag Description
compensationMethod	If the automatic payment feature is not configured for the account, the compensation method is set to **INVD** (i.e. Invoiced). However, if the automatic payment feature is configured for the account and the automatic payment creation date is earlier than the batch business date, the compensation method is set to **DEBTD** (i.e. Debited). And, if the automatic payment feature is configured for the account and the automatic payment creation date is later than the batch business date, the compensation method is set to **DDBT** (i.e. Delayed Debit).
debitAccount	Displays the international bank account number of the account from where the charges and taxes should be automatically debited. Usually, this account is different from the account which has availed the services.
delayedDebitDate	Displays the date when the charges and taxes are automatically debited.
settlementAdvice	Displays the message advising the customer about the settlement of charges and taxes which are due. **Note:** Here, a bill message whose bill message type is set to **S** appears in this tag.
accountBalanceCurrencyCode	Displays the currency in which the account balance is shown.
settlementCurrencyCode	Displays the currency in which the account is settled.
hostCurrencyCode	Displays the division's base currency which is used during currency conversion.
taxCalculationMethod	Displays the calculation method which indicates how the taxes are applied on the account. The valid value is **Method C**.
taxRegionCode	Indicates the tax region in which the account resides.
bankContactName	Displays the primary name of the account servicer's contact person. **Note:** The account servicer's contact person is an individual whom you can contact at the financial institution's location regarding problems of a business nature.
bankContactPhone	Displays the business or home phone number of the account servicer's contact person.
bankContactFax	Displays the business or home phone number of the account servicer's contact person.
bankContactEmail	Displays the e-mail address of the account servicer's contact person.
currencyTranslation	Displays the details of an exchange rate which is used in the bill.
originalCurrency	Displays the currency from which the amount is converted.
targetCurrency	Displays the currency to which the amount is converted.
translationValue	Displays the exchange rate.
basis	Displays the currency in which the exchange rate is defined.
service	Displays the details of the service (i.e. price item) offered by the financial institution.
bankServiceID	This tag appears within the following tags: **If this tag is used within...** — **Then the purpose of the bankServiceID tag is...** service — Displays the service (i.e. price item) offered to the customer. serviceDetail — Displays the service (i.e. price item) offered to the customer.

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Tag Description
serviceDescription	This tag appears within the following tags: **If this tag is used within...** — **Then the purpose of the serviceDescription tag is...** service — Displays the description of the price item. serviceDetail — Displays the description of the price item.
serviceType	Displays the parent price item to which the price item is associated.
volume	Displays the quantity of the service (i.e. price item) which is charged in the bill.
pricingCurrencyCode	Displays the details of the pricing which is used to the charge the customer.
unitPrice	Displays the price item pricing used per unit to calculate the charge in the pricing currency.
amount	This tag appears within the following tags: **If this tag is used within...** — **Then the purpose of the amount tag is...** unitPrice — Displays the rate per unit in the pricing currency. originalChargeSettlement — Displays the calculated charge in the invoice currency. originalCharge — Displays the sum of service charge lines in the bill excluding the service charge lines with a tax distribution code (i.e. a distribution code where the **TAXRATE** characteristic is defined). The amount is expressed in the invoice currency of the account. taxableServiceCharge — Displays the sum of service charge lines with the same tax distribution code (i.e. a distribution code where the **TAXRATE** characteristic is defined). The amount is expressed in the invoice currency of the account. This calculated amount is used as a source for calculating the tax. taxableServiceChargeHost — Displays the sum of service charge lines with the same tax distribution code (i.e. a distribution code where the **TAXRATE** characteristic is defined). The amount is expressed in the division's base currency. totalTaxableSvcChargeHostAmount — Displays the sum of all taxable service charge lines. The amount is expressed in the division's base currency. taxIdentifierTotalTaxAmount — Displays the sum of taxes charged due to the same tax distribution code. The amount is expressed in the division's base currency. totalTaxAmount — Displays the sum total of all taxes charged to the receiver (i.e. customer). The amount is expressed in the division's base currency. settlementAmount — Displays the sum total of all service charges and taxes billed to the receiver (i.e. customer). The amount is expressed in the invoice currency of the account. taxDueToRegion — Displays the sum total of all taxes charged to the receiver (i.e. customer). The amount is expressed in the invoice currency of the account. serviceAdjustmentAmt — Displays the adjustment amount in the invoice currency.

Copyright © 2024, Oracle and/or its affiliates.

<table>
<tr><th>Tag Name</th><th>Tag Description</th></tr>
<tr><td></td><td>
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the amount tag is...</th></tr>
<tr><td>newCharge</td><td>Displays the adjustment amount billed to the customer.</td></tr>
</table>
</td></tr>
<tr><td>currencyCode</td><td>This tag appears within the following tags:
<table>
<tr><th>If this tag is used within...</th><th>Then the purpose of the currencyCode tag is...</th></tr>
<tr><td>unitPrice</td><td>Indicates the currency in which the rate is defined.</td></tr>
<tr><td>originalChargeSettlement</td><td>Indicates the currency in which the amount is calculated.</td></tr>
<tr><td>originalCharge</td><td>Indicates the currency in which the amount is calculated.</td></tr>
<tr><td>taxableServiceCharge</td><td>Indicates the currency in which the amount is calculated.</td></tr>
<tr><td>taxableServiceChargeHost</td><td>Indicates the currency in which the amount is calculated.</td></tr>
<tr><td>totalTaxableSvcChargeHostAmount</td><td>Indicates the currency in which the amount is calculated.</td></tr>
<tr><td>taxIdentifierTotalTaxAmount</td><td>Indicates the currency in which the amount is calculated.</td></tr>
<tr><td>totalTaxAmount</td><td>Indicates the currency in which the amount is calculated.</td></tr>
<tr><td>settlementAmount</td><td>Indicates the currency in which the amount is calculated.</td></tr>
<tr><td>taxDueToRegion</td><td>Indicates the currency in which the amount is calculated.</td></tr>
<tr><td>serviceAdjustmentAmt</td><td>Indicates the currency in which the adjustment is created.</td></tr>
<tr><td>newCharge</td><td>Indicates the currency in which the adjustment is created.</td></tr>
</table>
</td></tr>
<tr><td>priceMethod</td><td>Indicates the type of price. The valid values are:

UPRC - Indicates that the price is a unit price.
FCHG - Indicates that the price is a flat price.
MCHG - Indicates that the price is a minimum price.
MXRD - Indicates that the price is a maximum price.
TIER - Indicates that the price is a step tiered price.
TPRC - Indicates that the price is a threshold tiered price.
</td></tr>
<tr><td>paymentMethod</td><td>Displays the type of charge created for the service. The valid values are:

FLAT - Indicates that the charge is an SQI-based billable charge.
PVCH - Indicates that the charge is a pass through memo charge.
INVS - Indicates that the charge is a pass through billable charge.
FREE - Indicates that the charge is a 100% discounted charge.
</td></tr>
<tr><td>originalChargeSettlement</td><td>Displays the details of the calculated charge.</td></tr>
<tr><td>taxDetails</td><td>Displays the details about the tax which is calculated on the charges.</td></tr>
<tr><td>taxRegion</td><td>Displays the details of the tax region.</td></tr>
<tr><td>taxRegionNumber</td><td>Displays the tax region in which the account resides.
Note: The tax region is derived from the TAXREG characteristic of the account.</td></tr>
</table>

Copyright © 2024, Oracle and/or its affiliates.

Tag Name	Tag Description
taxRegionName	Displays the tax region in which the account resides.
	Note: The tax region is derived from the **TAXREG** characteristic of the account.
customerTaxId	Displays the tax identification number of the receiver (i.e. financial institution or organization).
	Note: The tax ID is derived from the **TAXID** characteristic of the account.
taxInvoiceNumber	Displays the bill ID.
serviceDetail	Displays the basic details of the price item.
originalCharge	Displays the sum of service charge lines in the bill excluding the service charge lines with a tax distribution code.
taxCalculation	Displays the details of tax calculation.
hostCurCode	Displays the division's base currency which is used as the host currency.
taxHostConversion	Displays the sum of service charge lines with a tax distribution code. This tag is added multiple times in the statement when different tax distribution codes are used for the service.
taxableServiceCharge	Displays the sum of service charge lines with the same tax distribution code. The amount is expressed in the invoice currency.
taxableServiceChargeHost	Displays the sum of service charge lines with the same tax distribution code. The amount is expressed in the division's base currency.
totalTaxableSvcChargeHostAmount	Displays the sum of all taxable service charge lines in the division's base currency.
taxCalculationList	Displays the details of the tax applied on the service.
taxIdentificationNumber	Displays the distribution code where tax is applicable. In ORMB terms, it is a distribution code where the **TAXRATE** characteristic is defined.
taxIdentifierDescription	Displays the description of the distribution code.
taxIdentifierRate	Displays the rate at which the tax is calculated.
	Note: The tax rate is derived from the **TAXRATE** characteristic of the distribution code.
taxIdentifierTotalTaxAmount	Displays the sum of taxes charged due to the tax distribution code.
totalTaxAmount	Displays the sum total of all taxes charged to the receiver (i.e. customer).
settlementAmount	Displays the sum total of all service charges and taxes billed to the receiver (i.e. customer).
taxDueToRegion	Displays the sum total of all taxes charged to the receiver (i.e. customer).
serviceAdjustment	Displays the details of the adjustment which is created for the service and posted onto the bill.
Type	Displays the adjustment type using which the adjustment is created.
description	Displays the description of the adjustment type.
serviceAdjustmentAmt	Displays the adjustment amount.
adjustmentErrorDate	Displays the date when the adjustment is created.
serviceAdjustmentID	Displays the adjustment ID.
newCharge	Displays the adjustment amount billed to the customer.

Copyright © 2024, Oracle and/or its affiliates.

Related Topics

For more information on...	See...
TWIST Format	TWIST Format on page 587
Field Mapping for the TWIST Format	Field Mapping for the TWIST Format on page 602
Sample Statement in the TWIST Format	Sample Statement in the TWIST Format on page 637

Sample Statement in the CAMT Format

In the below sample statement, the following charges, adjustments, and taxes are billed to Green Corp. by Seven Bank:

Charge/ Adjustment	Price Method	Charge Rate (Per Unit)	Charge Amount	Adjustment Amount	Tax Rate (Percentage)	Tax Amount
Minimum Balance Charge (P001)	FCHG	$2	$2	-	1	$0.02
ATM Fees (P002)	FCHG	$5	$5	-	2	$0.1
Foreign Transaction Charge (P003)	TPRC	$0.2	$20	-	2	$0.4
Interest Earned	-	-	-	$10	-	-


```
<Document>
    <BkSvcsBllgStmt>
        <RptHdr>
            <RptId>5656587978</RptId>
        </RptHdr>
        <MsgPgntn>
            <PgNb></PgNb>
            <LastPgInd></LastPgInd>
        </MsgPgntn>
        <BllgStmtGrp>
            <GrpId>4578767989</GrpId>
            <Sndr>
                <Nm>Seven Bank</Nm>
                <LglNm>Seven Bank Limited</LglNm>
                <PstlAdr>
                    <AdrTp></AdrTp>
                    <Dept></Dept>
                    <SubDept></SubDept>
                    <StrtNm></StrtNm>
                    <BldgNb></BldgNb>
                    <PstCd>94122</PstCd>
                    <TwnNm>San Franciso</TwnNm>
                    <CtrySubDvsn>CA</CtrySubDvsn>
                    <Ctry>USA</Ctry>
                    <AdrLine>Saturn Park, Saturn Street, River Side</
AdrLine>
                    <Flr></Flr>
                    <PstBx></PstBx>
                    <BldgNm></BldgNm>
                    <Room></Room>
                </PstlAdr>
```


Copyright © 2024, Oracle and/or its affiliates.

```
                <Id>
                   <FIId>
                        <BICFI>76CA76CA</BICFI>
                    </FIId>
                </Id>
                <CtryOfRes>USA</CtryOfRes>
                <CtctDtls>
                    <NmPrfx>Ms</NmPrfx>
                    <Nm>G, LINDA</Nm>
                    <PhneNb>415-555-0103</PhneNb>
                    <MobNb>415-555-0106</MobNb>
                    <FaxNb>415-555-0107</FaxNb>
                    <EmailAdr>linda.g@example.com</EmailAdr>
                </CtctDtls>
            </Sndr>
            <SndrIndvCtct>
                <NmPrfx>Mr</NmPrfx>
                <Nm>G, RUSSEL</Nm>
                <PhneNb>415-555-0104</PhneNb>
                <MobNb>415-555-0105</MobNb>
                <FaxNb>415-555-0107</FaxNb>
                <EmailAdr>russel.g@example.com</EmailAdr>
            </SndrIndvCtct>
            <Rcvr>
                <Nm>Green Corp.</Nm>
                <LglNm>Green Corporation</LglNm>
                <PstlAdr>
                    <AdrTp>BILLING</AdrTp>
                    <Dept></Dept>
                    <SubDept></SubDept>
                    <StrtNm></StrtNm>
                    <BldgNb></BldgNb>
                    <PstCd>90020</PstCd>
                    <TwnNm>Los Angeles</TwnNm>
                    <CtrySubDvsn>CA</CtrySubDvsn>
                    <Ctry>USA</Ctry>
                    <AdrLine>Mars Park, Mars Street, Lake Side</AdrLine>
                    <Flr></Flr>
                    <PstBx></PstBx>
                    <BldgNm></BldgNm>
                    <Room></Room>
                </PstlAdr>
                <Id>
                    <OrgId>
                        <AnyBIC>34CA34CA</AnyBIC>
                    </OrgId>
                    <FIId>
                        <BICFI></BICFI>
                    </FIId>
                </Id>
                <CtryOfRes>USA</CtryOfRes>
                <CtctDtls>
                    <NmPrfx>Mr</NmPrfx>
                    <Nm>G, ALLEN</Nm>
                    <PhneNb>323-555-0111</PhneNb>
                    <MobNb>323-555-0112</MobNb>
                    <FaxNb>323-555-0113</FaxNb>
                    <EmailAdr>allen.g@example.com</EmailAdr>
                </CtctDtls>
            </Rcvr>
            <RcvrIndvCtct>
                <NmPrfx>Mr</NmPrfx>
                <Nm>G, JOHNSON</Nm>
                <PhneNb>323-555-0114</PhneNb>
```


Copyright © 2024, Oracle and/or its affiliates.

```
                <MobNb>323-555-0115</MobNb>
                <FaxNb>323-555-0113</FaxNb>
                <EmailAdr>johnson.g@example.com</EmailAdr>
            </RcvrIndvCtct>
            <BllgStmt>
                <StmtId>4578672301</StmtId>
                <FrToDt>
                    <FrDt>2020-01-01</FrDt>
                    <ToDt>2020-01-31</ToDt>
                </FrToDt>
                <CreDtTm>2020-02-01T13:57:07</CreDtTm>
                <Sts>ORGN</Sts>
                <AcctChrtcs>
                    <AcctLvl>DETL</AcctLvl>
                    <CshAcct>
                     <Id>
                            <IBAN>77AA77BB77</IBAN>
                        </Id>
                        <Tp>
                            <Cd>CC</Cd>
                            <Prtry>Classic Customer</Prtry>
                        </Tp>
                        <Ccy>USD</Ccy>
                    </CshAcct>
                    <AcctSvcr>
                        <FinInstnId>
                            <BICFI>75BA75BA</BICFI>
                        </FinInstnId>
                        <Nm>Seven Bank</Nm>
                        <PstlAdr>
                            <Dept></Dept>
                            <SubDept></SubDept>
                            <StrtNm></StrtNm>
                            <BldgNb></BldgNb>
                            <PstCd>94122</PstCd>
                            <TwnNm>San Franciso</TwnNm>
                            <CtrySubDvsn>CA</CtrySubDvsn>
                            <Ctry>USA</Ctry>
                            <AdrLine>Saturn Park, Saturn Street, River '
Side</AdrLine>
                        </PstlAdr>
                    </AcctSvcr>
                    <CompstnMtd>INVD</CompstnMtd>
                    <DbtAcct>
                        <IBAN></IBAN>
                    </DbtAcct>
                    <DelydDbtDt></DelydDbtDt>
                    <SttlmAdvc>Bank Settlement Advice</SttlmAdvc>
                    <AcctBalCcyCd>USD</AcctBalCcyCd>
                    <SttlmCcyCd>USD</SttlmCcyCd>
                    <HstCcyCd>USD</HstCcyCd>
                    <Tax>
                        <ClctnMtd>Method C</ClctnMtd>
                        <Rgn>SANFRAN</Rgn>
                    </Tax>
                    <AcctSvcrCtct>
                        <NmPrfx>Mr</NmPrfx>
                        <Nm>G, RUSSEL</Nm>
                        <PhneNb>415-555-0104</PhneNb>
                        <MobNb>415-555-0105</MobNb>
                        <FaxNb>415-555-0107</FaxNb>
                        <EmailAdr>russel.g@example.com</EmailAdr>
                    </AcctSvcrCtct>
                </AcctChrtcs>
```


Copyright © 2024, Oracle and/or its affiliates.

```
                <CcyXchg>
                    <SrcCcy></SrcCcy>
                    <TrgtCcy></TrgtCcy>
                    <XchgRate></XchgRate>
                    <UnitCcy></UnitCcy>
                    <QtnDt></QtnDt>
                </CcyXchg>
                <Svc>
                    <SvcDtl>
                        <BkSvc>
                            <Id>P001</Id>
                            <Desc>Minimum Balance Charge</Desc>
                            <SvcTp></SvcTp>
                        </BkSvc>
                        <Vol></Vol>
                    </SvcDtl>
                    <Pric>
                        <Ccy>USD</Ccy>
                        <UnitPric>
                            <Amt Ccy="USD">2</Amt>
                            <Sgn>TRUE</Sgn>
                        </UnitPric>
                        <Mtd>FCHG</Mtd>
                    </Pric>
                    <PmtMtd>INVS</PmtMtd>
                    <OrgnlChrgSttlmAmt>
                        <Amt Ccy="USD">2</Amt>
                        <Sgn>TRUE</Sgn>
                    </OrgnlChrgSttlmAmt>
                </Svc>
                <Svc>
                    <SvcDtl>
                        <BkSvc>
                            <Id>P002</Id>
                            <Desc>ATM Fees</Desc>
                            <SvcTp></SvcTp>
                        </BkSvc>
                        <Vol></Vol>
                    </SvcDtl>
                    <Pric>
                        <Ccy>USD</Ccy>
                        <UnitPric>
                            <Amt Ccy="USD">5</Amt>
                            <Sgn>TRUE</Sgn>
                        </UnitPric>
                        <Mtd>FCHG</Mtd>
                    </Pric>
                    <PmtMtd>INVS</PmtMtd>
                    <OrgnlChrgSttlmAmt>
                        <Amt Ccy="USD">5</Amt>
                        <Sgn>TRUE</Sgn>
                    </OrgnlChrgSttlmAmt>
                </Svc>
                <Svc>
                    <SvcDtl>
                        <BkSvc>
                            <Id>P003</Id>
                            <Desc>Foreign Transaction Charge</Desc>
                            <SvcTp></SvcTp>
                        </BkSvc>
                        <Vol>10000</Vol>
                    </SvcDtl>
                    <Pric>
                        <Ccy>USD</Ccy>
```

Copyright © 2024, Oracle and/or its affiliates.

```
                        <UnitPric>
                            <Amt Ccy="USD">0.2</Amt>
                            <Sgn>TRUE</Sgn>
                        </UnitPric>
                        <Mtd>TPRC</Mtd>
                    </Pric>
                    <PmtMtd>FLAT</PmtMtd>
                    <OrgnlChrgSttlmAmt>
                        <Amt Ccy="USD">20</Amt>
                        <Sgn>TRUE</Sgn>
                    </OrgnlChrgSttlmAmt>
                </Svc>
                <TaxRgn>
                    <RgnNb>D001</RgnNb>
                    <RgnNm>D001</RgnNm>
                    <CstmrTaxId>677699899</CstmrTaxId>
                    <InvcNb>8787879000</InvcNb>
                    <MtdC>
                        <SvcDtl>
                            <BkSvc>
                                <Id>P001</Id>
                                <Desc>Minimum Balance Charge</Desc>
                            </BkSvc>
                            <SvcChrgAmt>
                                <Amt></Amt>
                                <Sgn></Sgn>
                            </SvcChrgAmt>
                        </SvcDtl>
                        <SvcDtl>
                            <BkSvc>
                                <Id>P002</Id>
                                <Desc>ATM Fees</Desc>
                            </BkSvc>
                            <SvcChrgAmt>
                                <Amt></Amt>
                                <Sgn></Sgn>
                            </SvcChrgAmt>
                        </SvcDtl>
                        <SvcDtl>
                            <BkSvc>
                                <Id>P003</Id>
                                <Desc>Foreign Transaction Charge</Desc>
                            </BkSvc>
                            <SvcChrgAmt>
                                <Amt></Amt>
                                <Sgn></Sgn>
                            </SvcChrgAmt>
                        </SvcDtl>
                        <TaxClctn>
                            <HstCcy>USD</HstCcy>
                                <TaxblSvcChrgConvs>
                                    <SrcAmt>
                                        <Amt Ccy="USD">2</Amt>
                                        <Sgn>TRUE</Sgn>
                                    </SrcAmt>
                                    <HstAmt>
                                        <Amt Ccy="USD">2</Amt>
                                        <Sgn>TRUE</Sgn>
                                    </HstAmt>
                                </TaxblSvcChrgConvs>
                                <TaxblSvcChrgConvs>
                                    <SrcAmt>
                                        <Amt Ccy="USD">5</Amt>
                                        <Sgn>TRUE</Sgn>
```


Copyright © 2024, Oracle and/or its affiliates.

```
                                    </SrcAmt>
                                    <HstAmt>
                                        <Amt Ccy="USD">5</Amt>
                                        <Sgn>TRUE</Sgn>
                                    </HstAmt>
                                </TaxblSvcChrgConvs>
                                <TaxblSvcChrgConvs>
                                    <SrcAmt>
                                        <Amt Ccy="USD">20</Amt>
                                        <Sgn>TRUE</Sgn>
                                    </SrcAmt>
                                    <HstAmt>
                                        <Amt Ccy="USD">20</Amt>
                                        <Sgn>TRUE</Sgn>
                                    </HstAmt>
                                </TaxblSvcChrgConvs>
                                <TtlTaxblSvcChrgHstAmt>
                                    <Amt Ccy="USD">27</Amt>
                                    <Sgn>TRUE</Sgn>
                                </TtlTaxblSvcChrgHstAmt>
                                <TaxId>
                                    <Nb>D005</Nb>
                                    <Desc>D005</Desc>
                                    <Rate>1</Rate>
                                    <TtlTaxAmt>
                                        <Amt Ccy="USD">0.02</Amt>
                                        <Sgn>TRUE</Sgn>
                                    </TtlTaxAmt>
                                </TaxId>
                                <TaxId>
                                    <Nb>D006</Nb>
                                    <Desc>D006</Desc>
                                    <Rate>2</Rate>
                                    <TtlTaxAmt>
                                        <Amt Ccy="USD">0.1</Amt>
                                        <Sgn>TRUE</Sgn>
                                    </TtlTaxAmt>
                                </TaxId>
                                <TaxId>
                                    <Nb>D007</Nb>
                                    <Desc>D007</Desc>
                                    <Rate>2</Rate>
                                    <TtlTaxAmt>
                                        <Amt Ccy="USD">0.4</Amt>
                                        <Sgn>TRUE</Sgn>
                                    </TtlTaxAmt>
                                </TaxId>
                                <TtlTax>
                                    <Amt Ccy="USD">0.52</Amt>
                                    <Sgn>TRUE</Sgn>
                                </TtlTax>
                            </TaxClctn>
                        </MtdC>
                        <SttlmAmt>
                            <Amt Ccy="USD">27.52</Amt>
                            <Sgn>TRUE</Sgn>
                        </SttlmAmt>
                        <TaxDueToRgn>
                            <Amt Ccy="USD">0.52</Amt>
                            <Sgn>TRUE</Sgn>
                        </TaxDueToRgn>
                    </TaxRgn>
                    <SvcAdjstmnt>
                        <Tp>INTEARN</Tp>
```


Copyright © 2024, Oracle and/or its affiliates.

```
                    <Desc>Interest Earned</Desc>
                    <Amt>
                        <Amt Ccy="USD">10</Amt>
                        <Sgn>TRUE</Sgn>
                    </Amt>
                    <ErrDt>2020-01-31</ErrDt>
                    <AdjstmntId>5656787856</AdjstmntId>
                    <NewChrgAmt>
                        <Amt Ccy="USD">10</Amt>
                        <Sgn>TRUE</Sgn>
                    </NewChrgAmt>
                </SvcAdjstmnt>
            </BllgStmt>
        </BllgStmtGrp>
    </BkSvcsBllgStmt>
</Document>
```

Sample Statement in the TWIST Format

In the below sample statement, the following charges, adjustments, and taxes are billed to Green Corp. by Seven Bank:

Charge/ Adjustment	Price Method	Charge Rate (Per Unit)	Charge Amount	Adjustment Amount	Tax Rate (Percentage)	Tax Amount
Minimum Balance Charge (P001)	FCHG	$2	$2	-	1	$0.02
ATM Fees (P002)	FCHG	$5	$5	-	2	$0.1
Foreign Transaction Charge (P003)	TPRC	$0.2	$20	-	2	$0.4
Interest Earned	-	-	-	$10	-	-

```
<Twist>
    <header>9567523671</header>
    <electronicStatement>
        <statementHeader>
            <stmtSender>
                <name>Seven Bank</name>
                <legalName>Seven Bank Limited</legalName>
                <contactInfo>
                    <postAddress>
                        <addressIdentifier></addressIdentifier>
                        <departmentName></departmentName>
                        <streetName></streetName>
                        <buildingNum></buildingNum>
                        <postCode>94122</postCode>
                        <city>San Franciso</city>
                        <countrySubDivision>
                            <name>CA</name>
                        </countrySubDivision>
                        <country>USA</country>
                        <addressLine>Saturn Park, Saturn Street, River '
Side</addressLine>
                        <floor></floor>
                        <postBox></postBox>
```

Copyright © 2024, Oracle and/or its affiliates.

```
                        <buildingName></buildingName>
                        <room></room>
                    </postAddress>
                </contactInfo>
                <orgId>
                    <orgIdNum>76CA76CA</orgIdNum>
                </orgId>
                <contactInfo>
                    <individualContact>
                        <name>G, RUSSEL</name>
                    </individualContact>
                    <phone>415-555-0104</phone>
                    <fax>415-555-0107</fax>
                    <Email>russel.g@example.com</Email>
                </contactInfo>
            </stmtSender>
            <stmtReceiver>
                <name>Green Corp.</name>
                <legalName>Green Corporation</legalName>
                <contactInfo>
                    <postAddress>
                        <addressIdentifier></addressIdentifier>
                        <departmentName></departmentName>
                        <streetName></streetName>
                        <buildingNum></buildingNum>
                        <postCode>90020</postCode>
                        <city>Los Angeles</city>
                        <countrySubDivision>
                            <name>CA</name>
                        </countrySubDivision>
                        <country>USA</country>
                        <addressLine>Mars Park, Mars Street, Lake Side</
addressLine>
                        <floor></floor>
                        <postBox></postBox>
                        <buildingName></buildingName>
                        <room></room>
                    </postAddress>
                </contactInfo>
                <orgId>
                    <orgIdNum>34CA34CA</orgIdNum>
                </orgId>
                <contactInfo>
                    <individualContact>
                        <name>G, ALLEN</name>
                    </individualContact>
                    <phone>323-555-0111</phone>
                    <fax>323-555-0113</fax>
                    <Email>allen.g@example.com</Email>
                </contactInfo>
            </stmtReceiver>
        </statementHeader>
        <statement>
            <account>
                <statementStartDate>2020-01-01</statementStartDate>
                <statementEndDate>2020-01-31</statementEndDate>
                <statementProductionDate>2020-02-01T13:57:07</
statementProductionDate>
                <statementStatus>ORGN</statementStatus>
                <accountLevel>DETL</accountLevel>
                <iban>77AA77BB77</iban>
                <domicileBankIdentifier>75BA75BA</domicileBankIdentifier>
                <compensationMethod>INVD</compensationMethod>
                <debitAccount></debitAccount>
```


Copyright © 2024, Oracle and/or its affiliates.

```
                <delayedDebitDate></delayedDebitDate>
                <settlementAdvice>Bank Settlement Advice</settlementAdvice>
                <accountBalanceCurrencyCode>USD</accountBalanceCurrencyCode>
                <settlementCurrencyCode>USD</settlementCurrencyCode>
                <hostCurrencyCode>USD</hostCurrencyCode>
                <taxCalculationMethod>Method C</taxCalculationMethod>
                <taxRegionCode>SANFRAN</taxRegionCode>
                <bankContactName>G, RUSSEL</bankContactName>
                <bankContactPhone>415-555-0104</bankContactPhone>
                <bankContactFax>415-555-0107</bankContactFax>
                <bankContactEmail>russel.g@example.com</bankContactEmail>
            </account>
            <currencyTranslation>
                <originalCurrency></originalCurrency>
                <targetCurrency></targetCurrency>
                <translationValue></translationValue>
                <basis></basis>
            </currencyTranslation>
            <service>
                <bankServiceID>P001</bankServiceID>
                <serviceDescription>Minimum Balance Charge</
serviceDescription>
                <serviceType></serviceType>
                <volume></volume>
                <pricingCurrencyCode>USD</pricingCurrencyCode>
                <unitPrice>
                    <amount>2</amount>
                    <currencyCode>USD</currencyCode>
                </unitPrice>
                <priceMethod>FCHG</priceMethod>
                <paymentMethod>INVS</paymentMethod>
                <originalChargeSettlement>
                    <amount>2</amount>
                    <currencyCode>USD</currencyCode>
                </originalChargeSettlement>
            </service>
            <service>
                <bankServiceID>P002</bankServiceID>
                <serviceDescription>ATM Fees</serviceDescription>
                <serviceType></serviceType>
                <volume></volume>
                <pricingCurrencyCode>USD</pricingCurrencyCode>
                <unitPrice>
                    <amount>5</amount>
                    <currencyCode>USD</currencyCode>
                </unitPrice>
                <priceMethod>FCHG</priceMethod>
                <paymentMethod>INVS</paymentMethod>
                <originalChargeSettlement>
                    <amount>5</amount>
                    <currencyCode>USD</currencyCode>
                </originalChargeSettlement>
            </service>
            <service>
                <bankServiceID>P003</bankServiceID>
                <serviceDescription>Foreign Transaction Charge</
serviceDescription>
                <serviceType></serviceType>
                <volume>10000</volume>
                <pricingCurrencyCode>USD</pricingCurrencyCode>
                <unitPrice>
                    <amount>0.2</amount>
                    <currencyCode>USD</currencyCode>
                </unitPrice>
```


Copyright © 2024, Oracle and/or its affiliates.

```
                <priceMethod>TPRC</priceMethod>
                <paymentMethod>FLAT</paymentMethod>
                <originalChargeSettlement>
                    <amount>20</amount>
                    <currencyCode>USD</currencyCode>
                </originalChargeSettlement>
            </service>
            <taxDetails>
                <taxRegion>
                    <taxRegionNumber>D001</taxRegionNumber>
                    <taxRegionName>D001</taxRegionName>
                    <customerTaxId>677699899</customerTaxId>
                    <taxInvoiceNumber>8787879000</taxInvoiceNumber>
                    <serviceDetail>
                        <bankServiceId>P001</bankServiceId>
                        <serviceDescription>Minimum Balance Charge</
serviceDescription>
                        <originalCharge>
                             <amount></amount>
                             <currencyCode></currencyCode>
                        </originalCharge>
                    </serviceDetail>
                    <serviceDetail>
                        <bankServiceId>P002</bankServiceId>
                        <serviceDescription>ATM Fees</serviceDescription>
                        <originalCharge>
                             <amount></amount>
                             <currencyCode></currencyCode>
                        </originalCharge>
                    </serviceDetail>
                    <serviceDetail>
                        <bankServiceId>P003</bankServiceId>
                        <serviceDescription>Foreign Transaction Charge</
serviceDescription>
                        <originalCharge>
                             <amount></amount>
                             <currencyCode></currencyCode>
                        </originalCharge>
                    </serviceDetail>
                    <taxCalculation>
                        <hostCurCode>USD</hostCurCode>
                        <taxHostConversion>
                            <taxableServiceCharge>
                                <amount>2</amount>
                                <currencyCode>USD</currencyCode>
                            </taxableServiceCharge>
                            <taxableServiceChargeHost>
                                <amount>2</amount>
                                <currencyCode>USD</currencyCode>
                            </taxableServiceChargeHost>
                        </taxHostConversion>
                        <taxHostConversion>
                            <taxableServiceCharge>
                                <amount>5</amount>
                                <currencyCode>USD</currencyCode>
                            </taxableServiceCharge>
                            <taxableServiceChargeHost>
                                <amount>5</amount>
                                <currencyCode>USD</currencyCode>
                            </taxableServiceChargeHost>
                        </taxHostConversion>
                        <taxHostConversion>
                            <taxableServiceCharge>
                                <amount>20</amount>
```


Copyright © 2024, Oracle and/or its affiliates.

```
                                <currencyCode>USD</currencyCode>
                            </taxableServiceCharge>
                            <taxableServiceChargeHost>
                                <amount>20</amount>
                                <currencyCode>USD</currencyCode>
                            </taxableServiceChargeHost>
                        </taxHostConversion>
                        <totalTaxableSvcChargeHostAmount>
                            <amount>27</amount>
                            <currencyCode>USD</currencyCode>
                        </totalTaxableSvcChargeHostAmount>
                        <taxCalculationList>
                            <taxIdentificationNumber>D005</
taxIdentificationNumber>
                            <taxIdentifierDescription>D005</
taxIdentifierDescription>
                            <taxIdentifierRate>1</taxIdentifierRate>
                            <taxIdentifierTotalTaxAmount>
                                <amount>0.02</amount>
                                <currencyCode>USD</currencyCode>
                            </taxIdentifierTotalTaxAmount>
                        </taxCalculationList>
                        <taxCalculationList>
                            <taxIdentificationNumber>D006</
taxIdentificationNumber>
                            <taxIdentifierDescription>D006</
taxIdentifierDescription>
                            <taxIdentifierRate>2</taxIdentifierRate>
                            <taxIdentifierTotalTaxAmount>
                                <amount>0.1</amount>
                                <currencyCode>USD</currencyCode>
                            </taxIdentifierTotalTaxAmount>
                        </taxCalculationList>
                        <taxCalculationList>
                            <taxIdentificationNumber>D007</
taxIdentificationNumber>
                            <taxIdentifierDescription>D007</
taxIdentifierDescription>
                            <taxIdentifierRate>2</taxIdentifierRate>
                            <taxIdentifierTotalTaxAmount>
                                <amount>0.4</amount>
                                <currencyCode>USD</currencyCode>
                            </taxIdentifierTotalTaxAmount>
                        </taxCalculationList>
                        <totalTaxAmount>
                            <amount>0.52</amount>
                            <currencyCode>USD</currencyCode>
                        </totalTaxAmount>
                    </taxCalculation>
                    <settlementAmount>
                        <amount>27.52</amount>
                        <currencyCode>USD</currencyCode>
                    </settlementAmount>
                    <taxDueToRegion>
                        <amount>0.52</amount>
                        <currencyCode>USD</currencyCode>
                    </taxDueToRegion>
                </taxRegion>
            </taxDetails>
            <serviceAdjustment>
                <Type>INTEARN</Type>
                <description>Interest Earned</description>
                <serviceAdjustmentAmt>
                    <amount>10</amount>
```


Copyright © 2024, Oracle and/or its affiliates.

```
                    <currency Code>USD</currencyCode>
                </serviceAdjustmentAmt>
                <adjustmentErrorDate>2020-01-31</adjustmentErrorDate>
                <serviceAdjustmentID>5656787856</serviceAdjustmentID>
                <newCharge>
                    <amount>10</amount>
                    <currencyCode>USD</currencyCode>
                </newCharge>
            </serviceAdjustment>
        </statement>
    </electronicStatement>
</Twist>
```

Statement Cycles

The accounts and contracts linked to a statement construct may have different bill cycles and as a result, their bills are produced on different dates. So when should a statement containing the consolidation of all this financial information be produced? To answer this question, there is a statement cycle.

Just as with a bill cycle, the statement cycle has a schedule. A statement cycle's schedule controls when a statement should be produced. Each statement construct will point to its own statement cycle.

Fastpath: Refer to Statement Construct Maintenance and Statement Cycle for more information.

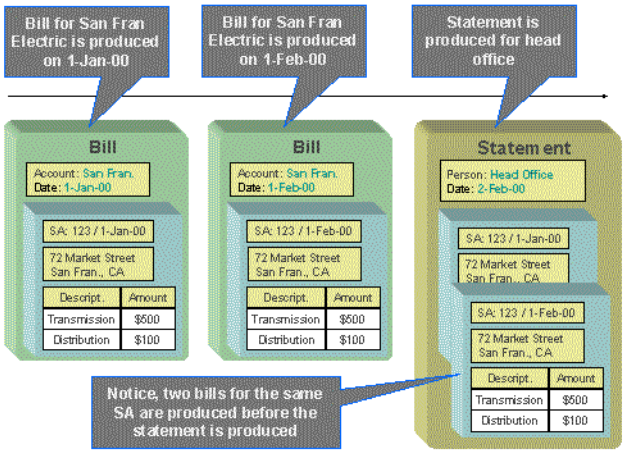
Create Statements Background Process

This statement construction background process (known by the batch control ID of `STMPRD`) is responsible for periodically creating statements. It works as follows:

- All statement cycles are selected with `pending` schedules whose production date is on or before the business date.
- For each such cycle, all of its statement constructs are selected. For each statement construct,
- - If a statement already exists for the cycle schedule and construct, it is deleted (this allows for reproduction after a mess up). Note well, this is true even if the statement is printed as you may have only spotted the problem after viewing the printed statement.
  - A statement is created.
  - All contract's associated with the construct details effective on the business date are extracted.
  - All "contract snapshots" associated with bills whose BILL DATE is within the construct detail's start and end dates that are not already linked to one of the construct's statements are linked to the new statement. Note, an "contract snapshot" is created for every contract linked to an account when a bill is completed for an account.
  - If no "contract snapshots" are found, the statement is removed.

Note: The following diagram illustrates how the above logic could result in multiple bills for a single contract being linked to the same statement.

Copyright © 2024, Oracle and/or its affiliates.



Fastpath: Refer to Statement Maintenance for more information about a statement and its details.

Note: This process is rerunnable. If a problem occurred during the statement creation, simply run the background process again. It will clean up all existing statements for each construct in the current cycle schedule and start again.

Fastpath: Refer to Printing Statements for more information for how a separate background process uses the information snapshot on a statement to construct the extract file used to print the statements.

Statement Routing

Just as with bills, statements require routing information in order for the system to understand how and where to send the statements. For example, the system needs to know the method of routing, the number of copies and the format of the statement. This information is captured on a statement construct.

Where Are Statements Sent

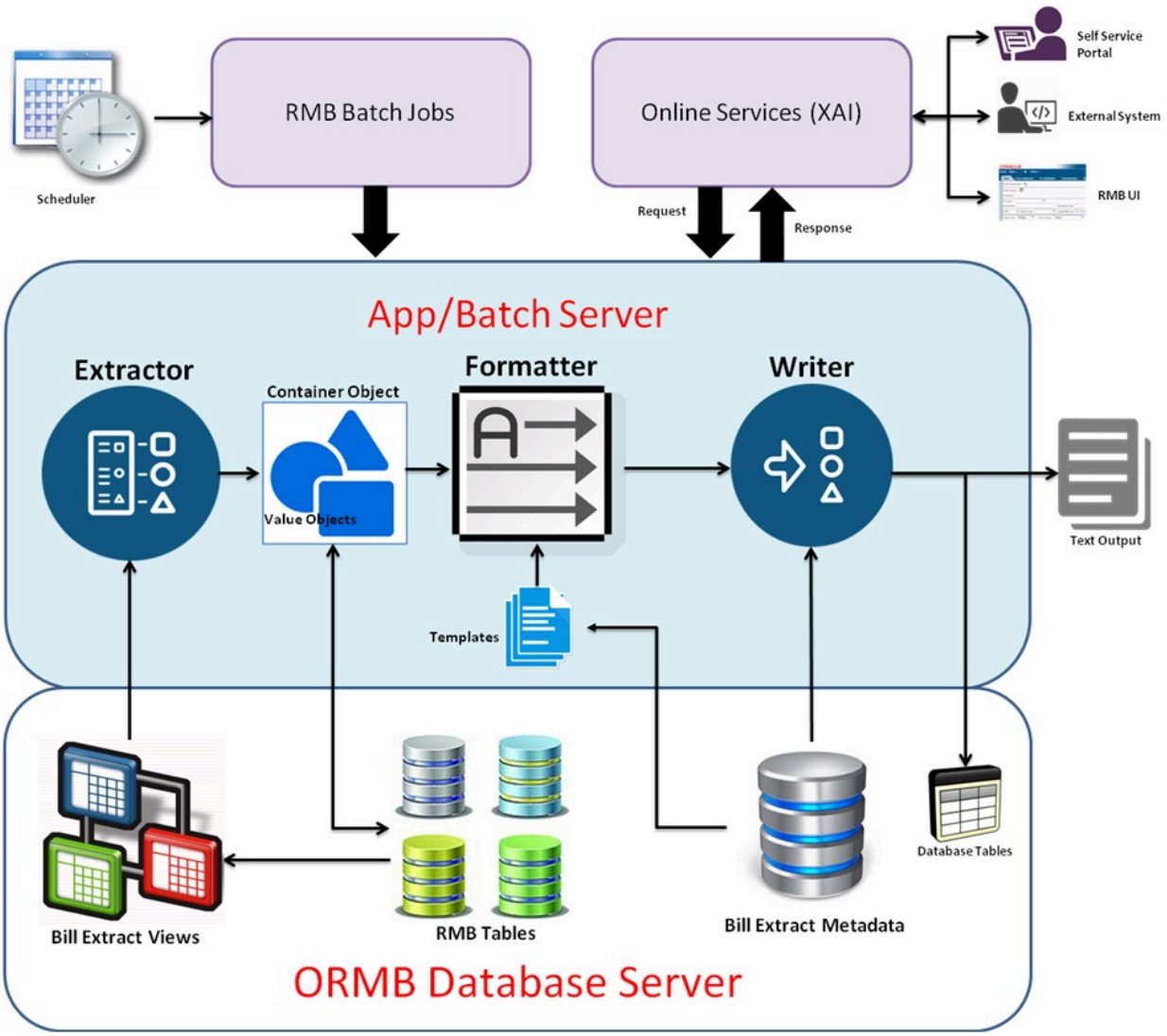
The system extracts a statement's address as per its statement construct's address source. Note, if the address source is `Person` and the person has a seasonal address effective on the business date, the seasonal address will be used.

If you need to reprint a statement because it was addressed incorrectly, correct the person's address, and then reprint the statement (by pressing the **Reprint** button on the statement). When you press this button, the statement will be marked for downloading. When your download process next runs, it simply extracts the statement construct's current address.

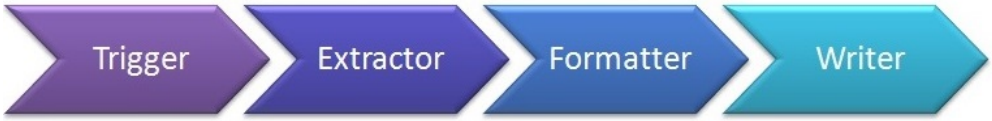
Statement Generation Process

Oracle Revenue Management and Billing enables you to generate statements for a person through the web service and batch control. The following figure graphically represents how the system extracts the details from the bills and generates a statement:

Copyright © 2024, Oracle and/or its affiliates.



The following figure indicates the different components involved in the statement generation process:



The major components involved in the statement generation process are:

- **Triggers** - The two main invocation contexts for the statement generation process are RMB Batch Jobs and Online Web Services. Typically, the batch jobs trigger fetching of records (data) for all accounts since the previous trigger, whereas the online service triggers fetching of data for one applicable account.
- **Extractor** - The main function of Extractor is to fetch data from the RMB database tables using the statement views and convert them into generic objects called **value objects**.
- **Formatter** - The Formatter converts the value objects into the desired format using the predefined templates.
- **Output Writer** - The Output Writer routes the formatted data to the database tables.

Note: The **Extractor**, **Formatter**, and **Output Writer** reside within the application or batch server, whereas the statement views, statement metadata, and tables reside within the database server.

The statement generation process receives its triggers through RMB Batch Jobs or Online Web Services. An external scheduler typically invokes the batch job at a desired frequency and extracts bill data for all those statement constructs

Copyright © 2024, Oracle and/or its affiliates.

whose statement cycle's processing date (which is earlier than the batch business date) is in the **Pending** status. On the other hand, the online web service typically triggers extract of an account's bill details and there will be appropriate authentication mechanism for triggering the web service. Once the statement generation process receives a trigger, either from a batch job or from an online service, the Extractor fetches the accounts' bill details from the database through the statement views. The Extractor then converts the data into generic objects called **value objects**, and passes them onto the Formatter.

The Formatter first identifies the format applicable for the account and fetches the corresponding template from the statement metadata table. The Formatter then merges the value objects and the fetched template to generate a text output, which is then passed on to the Output Writer. Based on the metadata configurations, the Output Writer writes the output to the database tables.

On-line Statement Production

Statements may be produced on-line as well as in batch. To produce a statement on-line, simply go to the Statement page, select the correct Construct ID and choose the Generate button.

This statement will include the financial information related to bills for the accounts and contracts linked to this construct, with the following criteria:

- The bills do not already appear on a statement
- The bill dates fall within the "effective period" for the statement construct. In other words, the bill falls within the start and end period for the statement construct details.

Fastpath: Refer to Statement Maintenance for more information.

How to Generate a Statement

You can generate a statement from the user interface or through a background process. For more information on how to generate a statement from the user interface, see On-line Statement Production on page 645.

To create statements in the background process, you need to:

If you want to...	Then...
Extract Bills in the Custom Format	Set the batch control type of the **Create Statements (STMPRD)** batch to **Timed** and define the following attributes: • Time Interval • Timer Active • User ID • Batch Language • Email Address
Extract Bills in the CAMT or TWIST Format	Set the batch control type of the **Create Statements in CAMT or TWIST Format (CISTMEXT)** batch to **Timed** and define the following attributes: • Time Interval • Timer Active • User ID • Batch Language • Email Address

For more information about these batches, see *Oracle Revenue Management and Billing Batch Guide*.

How to Regenerate a Statement

There might be situations when a bill is rebilled or canceled after generating the statement. In such scenarios, you can use the **C1_RegenerateStatement** inbound web service to regenerate a statement. While regenerating a statement, the

Copyright © 2024, Oracle and/or its affiliates.

inbound web service extracts the latest data from the bills. There is appropriate authentication mechanism for calling the service. It can be invoked by an external user application. You can specify the following input parameters in the inbound web service:

- Statement ID

On processing the request, the inbound web service returns the statement in the appropriate format. For more information about the tags, see XML Format Required for C1_RegenerateStatement on page 647.

Alternatively, you can regenerate a statement using up-to-date information from the user interface. This is possible by deleting the statement and then generating a new statement.

Note: Statements may be deleted, even if they have already been printed. This enables you to delete and regenerate statements.

Fastpath: Refer to Statement Maintenance for more information.

How to Search for a Statement

The **C1_SearchStatement** inbound web service enables you to search for a statement using various search criteria. It also allows you to search for the last 'n' statements of a particular account. There is appropriate authentication mechanism for calling the service. It can be invoked by an external user application. You can specify the following input parameters in the inbound web service:

- Last ‘n’ count
- From Date
- To Date
- Bill ID
- Account ID
- Statement ID

On processing the request, the inbound web service returns one or more statement IDs. For more information about the tags, see XML Format Required for C1_SearchStatement on page 648.

How to View a Statement

The **C1_ReprintStatement** inbound web service enables you to view and reprint a statement that is already generated. There is appropriate authentication mechanism for calling the service. It can be invoked by an external user application. You can specify the following input parameters in the inbound web service:

- Statement ID

On processing the request, the inbound web service returns the statement in the appropriate format. For more information about the tags, see XML Format Required for C1_ReprintStatement on page 649.

How to Download a Statement

Through the **Download Statements (STMDWLD)** batch, the system enables you to download the data extracted for a statement in a flat file. This flat file can be interfaced to your bill print software, or whatever mechanism you want to use for sending statements to the appropriate persons. The information that is extracted and placed on the flat file for each statement is controlled by the respective statement construct's statement route type's extract algorithm. For more information about the batch, see *Oracle Revenue Management and Billing Batch Guide*.

How to Archive a Statement

There might be situations when you may want to archive the data extracted for the statements in the database. In such scenarios, you can use the **Archive Statements (CI_ARCST)** batch. For more information about the batch, see *Oracle Revenue Management and Billing Batch Guide*.

Copyright © 2024, Oracle and/or its affiliates.

C1_RegenerateStatement Inbound Web Service Schema

The following code snippet illustrates how the tags are nested in the C1_RegenerateStatement schema:

```
<schema name="schema">
    <statementId mapField="STATEMENT_ID"/>
    <stmObj mapField="STM_OBJ"/>
</schema>
```

C1_SearchStatement Inbound Web Service Schema

The following code snippet illustrates how the tags are nested in the C1_SearchStatement schema:

```
<schema name="schema">
    <recordCount dataType="number" mapField="RECORDS_COUNT"/>
    <toDate dataType="date" mapField="TO_DT"/>
    <fromDate dataType="date" mapField="FROM_DT"/>
    <billId mapField="BILL_ID"/>
    <acctId mapField="ACCT_ID"/>
    <statementId mapField="STATEMENT_ID"/>
    <stmList type="list" mapList="STM_LIST">
        <statementId mapField="STATEMENT_ID"/>
    </stmList>
</schema>
```

C1_ReprintStatement Inbound Web Service Schema

The following code snippet illustrates how the tags are nested in the C1_ReprintStatement schema:

```
<schema name="schema">
    <statementId mapField="STATEMENT_ID"/>
    <stmObj mapField="STM_OBJ"/>
</schema>
```

XML Format Required for C1_RegenerateStatement

Note: We recommend you to refer the **C1_RegenerateStatement** inbound web service schema in parallel while understanding the below mentioned tags. This will help you to understand how the tags are nested in the schema.

Before calling the **C1_RegenerateStatement** inbound web service, you need to ensure that the XML file contains the following tags:

Tag Name	Tag Description	Mandatory (Yes or No)
schema	Used to specify the tags in an XML document.	Yes
statementId	Used to specify the statement ID.	Yes

The following table lists and describes the tags which are available in the **C1_RegenerateStatement** schema, but should not be included in the XML file:

Tag Name	Tag Description
stmObj	Displays the statement in the appropriate format.

Related Topic

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
C1_RegenerateStatement Inbound Web Service Schema	C1_RegenerateStatement Inbound Web Service Schema on page 647

XML Format Required for C1_SearchStatement

Note: We recommend you to refer the **C1_SearchStatement** inbound web service schema in parallel while understanding the below mentioned tags. This will help you to understand how the tags are nested in the schema.

Before calling the **C1_SearchStatement** inbound web service, you need to ensure that the XML file contains the following tags:

Tag Name	Tag Description	Mandatory (Yes or No)
schema	Used to specify the tags in an XML document.	Yes
recordCount	Used when you want to search last 'n' number of statements of an account.	Yes (Conditional) **Note:** Either the count or date range is required when the account ID is specified as a search criteria.
toDate	Used to search for statements which are created till a particular date.	Yes (Conditional) **Note:** Either the count or date range is required when the account ID is specified as a search criteria.
fromDate	Used to search for statements which are created from a particular date.	Yes (Conditional) **Note:** Either the count or date range is required when the account ID is specified as a search criteria.
billId	Used to search for statements which contains the data from a particular bill.	No
acctId	Used to search for statements which are created for a particular account.	No
statementId	Used to search for a particular statement.	No

Note: The bill ID, account ID, or the statement ID should be specified while searching for statements.

The following table lists and describes the tags which are available in the **C1_SearchStatement** schema, but should not be included in the XML file:

Tag Name	Tag Description
stmList	Lists the statements that meet the specified criteria.
statementId	Displays the statement ID.

Related Topic

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
C1_SearchStatement Inbound Web Service Schema	C1_SearchStatement Inbound Web Service Schema on page 647

XML Format Required for C1_ReprintStatement

Note: We recommend you to refer the **C1_ReprintStatement** inbound web service schema in parallel while understanding the below mentioned tags. This will help you to understand how the tags are nested in the schema.

Before calling the **C1_ReprintStatement** inbound web service, you need to ensure that the XML file contains the following tags:

Tag Name	Tag Description	Mandatory (Yes or No)
schema	Used to specify the tags in an XML document.	Yes
statementId	Used to specify the statement ID.	Yes

The following table lists and describes the tags which are available in the **C1_ReprintStatement** schema, but should not be included in the XML file:

Tag Name	Tag Description
stmObj	Displays the statement in the appropriate format.

Related Topic

For more information on...	See...
C1_ReprintStatement Inbound Web Service Schema	C1_ReprintStatement Inbound Web Service Schema on page 647

Printing Statements

The contents of this section describe the technical implementation of printing statements.

- Statement Route Types Control The Information Merged Onto Statements on page 649
- Technical Implementation Of Online Statement Image on page 650
- Technical Implementation Of Printing Statements In Batch on page 650
- Reproducing The Statement Print Flat File on page 650
- How To Reprint A Specific Statement on page 650
- The Doc 1 Records on page 650

Statement Route Types Control The Information Merged Onto Statements

Every statement references a statement route type. The route type controls the following statement print functions:

- It contains an algorithm that is responsible for extracting the information merged onto your statements. Specifically, algorithms of this type create the "flat file records" that are passed to your statement print software. Algorithms of this type are called under the following scenarios:
-
  - The background process that builds the flat file that's passed to your statement print software calls these algorithms to construct the "flat file records" for each statement.
  - If your statement print software has the ability to construct a real-time image of a statement (in a PDF), you can plug-in an `Online Statement Image` algorithm on the Installation Record. This type of algorithm will call the statement's dispatch group extract algorithm to extract the information that is merged onto the statement. Refer to Technical Implementation Of Online Statement Image for the details.

Copyright © 2024, Oracle and/or its affiliates.

- Refer to The Doc 1 Records for examples of the record layouts passed to the Doc 1 print software.
- It contains the ID of the background process that builds the flat file that's passed to your statement print software. The base package example of this process (known by the batch control ID of STMDWLD) simply calls each statement's construct's route type's extract algorithm to format the information placed onto the flat file.

Technical Implementation Of Online Statement Image

Users can view an image of any statement if you setup the following:

- Plug-in an `Online Statement Image` construction algorithm on the Installation Record. Refer to ONSD-ST for an example of such an algorithm. Note, if your statement print software is not capable of producing a PDF containing an image of a statement, users will not be able to view images of statements.
- Plug-in the appropriate extract algorithm on each statement route type. Algorithms of this type format the records that contain the information that is passed to your printing software. Refer to STEX-TX for an example of such an algorithm.

When you plug-in these algorithms, a button appears on Statement - Main. When a user presses this button, the following takes place:

- The installation record's `Online Statement Image` construction algorithm is executed.
- This algorithm calls the statement's route type's extract algorithm. This algorithm constructs the information that's merged onto the statement and returns it to the `Online Statement Image` algorithm. This algorithm, in turn, passes it to your statement print software.
- Your statement print software renders an image of the statement in a PDF and returns it to the `Online Statement Image` algorithm.
- And finally, the `Online Statement Image` algorithm displays the PDF in a separate Adobe session.

Technical Implementation Of Printing Statements In Batch

The batch process that extracts statement information (known by the batch control ID of STMDWLD) reads all statements in a given run number that are marked with its batch control ID. For each statement, it creates numerous records on a flat file. These records contain the information that is merged onto your statements.

The base package example of this process (known by the batch control ID of *STMDWLD*) simply calls the extract algorithm on the statement route type to format the information placed onto the flat file. Refer to Statement Route Types Control The Information On Statements for more information.

> **Note: If your software doesn't support online statement images.** The algorithm that formats statement extract records that's plugged in on statement route type serves two purposes: 1) it formats the records used to construct online images of a statement, and 2) it formats the records downloaded to your statement print software in batch. If your statement print software does not support the rendering of statement images real time, there is no need to create an extract algorithm. Rather, you should simply develop your own download process that both formats the extract records and downloads them (and then specify this batch process on your statement route type).

Reproducing The Statement Print Flat File

You can reproduce the flat file at any time. Simply request the STMDWLD process and specify the run number associated with the historic run.

How To Reprint A Specific Statement

If you need to reprint a specific statement, navigate to Statement - Main and press the **Reprint** button.

If your implementation has enabled the online creation of statement images, you can also press the **Display Statement** button on this page and then print the resultant PDF on your local printer.

The Doc 1 Records

Numerous different types of records are interfaced to the Doc 1 software. However, they all share a common structure:

Copyright © 2024, Oracle and/or its affiliates.

- The first 4 bytes are called the **Printer Key**. This field is a record type used by the Doc 1 statement template.
- The next 186 bytes are called the **Sort Key**. This field is used to ensure the various records used to construct a printed statement appear in the correct order in the interface file.
- The next 12 bytes are called the **Mailing Key**. This field is used to control the order in which the formatted statements are printed.
- The remaining bytes contain the **Statement Information** that appears on the printed statement. The type of information differs for each type of record.

The topics in this section describe each component.

Printer Keys (Record Types)

The following table describes the different types of records in the statement record:

Record Type	Description	Notes
0010	Global extract information	Required. 1 per flat file.
0100	Statement record	Required. 1 per statement.
0120	Total by currency record	Required. At least one per statement.
0500	Construct detail record	Required. At least one per construct detail linked to the statement construct.
0900	Contract record	1 per contract
1100	Bill record	1 per bill
1300	Financial Transaction (FT) record	1 per FT linked to the contract.
3600	End of record type 1100	1 for each bill
3700	End of record type 0900	1 for each contract
5000	End of record type 0500	1 for each statement construct detail
9999	End of record type 0100	1 for each Statement

Sort Key

The following table describes the structure of the sort key that appears on the print record of each statement:

Field Name	Format	Record Type	Source/Value/Description
Statement ID	A12	All	Prime key of the statement
Sequence Number	N2	All	Set to "1"
Copy Number	N2	All	Defaulted to "1" on the first copy, set to "1" when a statement is requested on-line
Statement record group	A2	0010	Blank
		0100	10
		0120	20
		0500, 0900, 1100, 1300, 3600, 3700, 5000	30
		9999	60
Construct detail sort sequence 1	N3	0010, 0100, 0120, 9999	blank

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Format	Record Type	Source/Value/Description
		0500, 0900, 1100, 1300, 3600, 3700, 5000	Statement construct's PRT_SEQ
Construct detail sort sequence 2	A10	0010, 0100, 0120, 9999	blank
		0500, 0900, 1100, 1300, 3600, 3700, 5000	Statement construct's STM_CNST_DTL_ID
Contract Print Priority	A2	0010, 0100, 0120, 0500, 5000, 9999	Blank
		0900, 1100, 1300, 3600, 3700	The contract's contract type's print priority
Contract Id	A10	0010, 0100, 0120, 0500, 5000, 9999	Blank
		0900, 1100, 1300, 3600, 3700	The contract's contract ID
Contract record group	A2	0010, 0100, 0120, 0500, 5000, 9999	Blank
		0900	10
		1100, 1300, 3600	20
		3700	30
Bill Date	A10	0010, 0100, 0120, 0500, 0900, 3700, 5000, 9999	Blank
		1100, 1300, 3600	Bill's bill date in the format YYYYMMDD.
Bill Id	A12	0010, 0100, 0120, 0500, 0900, 3700, 5000, 9999	Blank
		1100, 1300, 3600	Bill's bill id
Bill record group	A2	0010, 0100, 0120, 0500, 0900, 3700, 5000, 9999	Blank
		1100	10
		1300	20
		3600	30

Mailing Key

The following table describes the structure of the mailing key that appears on each statement print record:

Field Name	Format	Source/Value/Description
Postal code	A12	This is the postal code on address associated with the statement.

Statement Information

The topics in this section describe the information that appears on each on each statement print record.

Contents

Copyright © 2024, Oracle and/or its affiliates.

Global Extract Information Record (0010)

The following table lists the header elements that contain information about the extract run:

Field Name	Format	Source/Value/Description
BATCH_CD	A8	Batch Control Code of the extract process.
BATCH_NBR	N10	Batch Number of the extract process
BATCH_RERUN_NBR	N10	Batch Rerun number of the extract process.
STATEMENT_CNT	N10	Count of statements on this extract. Multiple copies of a statement are counted multiple as well.
EXTRACT_DTTM	A31	System time of extraction. Formatted according to user profile.
STMT_ID_START	A12	This is only used when a statement is produced as a result of an online request (from Statement - Main). It contains the Statement Id to extract.
STMT_ID_END	A12	This is only used when a statement is produced as a result of an online request (from Statement - Main). It contains the Statement Id to extract.
THREAD_NBR	N10	This field is only used if statements are extracted in batch. It contains the thread number in which the statement was extracted.
THREAD_CNT	N10	This field is only used if statements are extracted in batch. It contains the total number of threads that were extracted.
STATEMENT_IND	A1	Defaulted to "Y"

Statement Record (0100)

The following table describes the fields that appear in the statement record:

Field Name	Format	Source/Value/Description
STMT_ID	A12	CI_STM
PER_ID	A10	CI_STM_CNST
STMT_DT	A31	STM_DT from CI_STM. Formatted according to user profile.
COPY_NBR	N1	"1"
NBR_STMT_COPIES	N1	"1"
ENTITY_NAME	A64	Main name of CI_PER (from CI_PER_NAME).
ADDRESS_SBR	A361	Address where the statement should be sent (the address is defined on the statement's statement construct).
ENTITY_NAME1	A64	CI_PER
ENTITY_NAME2	A64	CI_PER
ENTITY_NAME3	A64	CI_PER
DESCR50	A50	CI_STM_CNST

Total By Currency Record (0120)

One total record will be created for each currency code associated with the FT's associated with the contract snapshots linked to the statement.

Copyright © 2024, Oracle and/or its affiliates.

The following table describes the fields that appear in the total by currency record:

Field Name	Format	Source/Value/Description
CURRENCY_CD	A3	This is the currency code associated with the summary information.
CUR_BAL	A30	This is the sum of CUR_AMT snapshot on the contract snapshots (CI_BILL_SA) linked to the statement. Contract snapshots (CI_BILL_SAs) are referenced on CI_STM_DTL.
BILL_AMT	A30	This is the sum of CUR_AMT from CI_FTs classified as "corrections" and "current charges" on bill's associated with the statement. Please note, only those FT's associated with contracts linked to the statement will be amalgamated. This is formatted according to user's display profile.
CUR_ADJ_AMT	A30	This is the sum of CUR_AMT from CI_FTs classified as "adjustments" on bills associated with the statement. Please note, only those FT's associated with contracts linked to the statement will be amalgamated. This is formatted according to user's display profile.
CUR_PAY_AMT	A30	This is the sum of CUR_AMT from CI_FTs classified as "payments" on bills associated with the statement. Please note, only those FT's associated with contracts linked to the statement will be amalgamated. This is formatted according to user's display profile.

Statement Construct Detail Record (0500)

The following table describes the fields that appear in the statement construct detail record:

Field Name	Format	Source/Value/Description
STM_CNST_ID	A12	CI_STM_DTL
Contract / Account Indicator	A1	A switch to indicate if the statement construct is for a contract of an account. 'S' for contract, 'A' for account.
Unique ID	A10	The unique identifier of the account or contract
Contract / Account Info	A160	Contains the standard information about the account or contract
STMT_PRT_DESCR	A50	CI_STM_CNST

Contract Record (0900)

The following table describes the fields that appear in the contract record:

Field Name	Format	Source/Value/Description
Address Information	A120	This is the standard format of an address that is displayed throughout the system.
Contract Information	A120	This is the standard contract information that is displayed throughout the system
SVC_TYPE_CD	A2	CI_SA_TYPE

Bill Record (1100)

The following table describes the fields that appear in the bill record:

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
BILL_ID	A12	
Completion Date	A31	Date formatted according to user profile.
Due Date	A31	Date formatted according to user profile.
Contract Ending Balance	A30	
FT Exist Indicator	A1	

Financial Transaction (FT) Record (1300)

The following table describes the fields that appear in the financial transaction record:

Field Name	Format	Source/Value/Description
FT_ID	A12	CI_FT
ARS_DT	A31	CI_FT (The date when the transaction starts aging). Date formatted according to user profile.
AMOUNT	A30	CUR_AMT from CI_FT formatted according to user display profile.
FT_DESCR	A120	Formatted FT information.

End Bill Record (3600)

The following table describes the fields that appear at the end of the bill record:

Field Name	Format	Source/Value/Description
Dummy field		

End Contract Record (3700)

The following table describes the fields that appear at the end of the contract record:

Field Name	Format	Source/Value/Description
Dummy field		

End Statement Construct Detail Record (5000)

The following table describes the fields that appear at the end of the statement construct detail record:

Field Name	Format	Source/Value/Description
Dummy field		

End Statement Record (9999)

The following table describes the fields that appear at the end of the statement record:

Field Name	Format	Source/Value/Description
Dummy field		

Address Sub-record

The following table describes the fields that appear in the address record:

Copyright © 2024, Oracle and/or its affiliates.

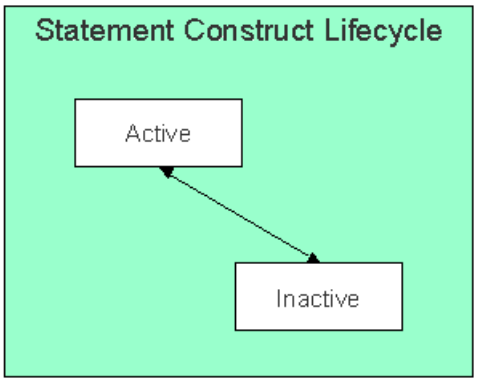
Field Name	Format	Description
COUNTRY	A3	
ADDRESS1	A64	
ADDRESS2	A64	
ADDRESS3	A64	
ADDRESS4	A64	
CITY	A30	
NUM1	A6	
NUM2	A4	
HOUSE_TYPE	A2	
COUNTY	A30	
STATE	A6	
POSTAL	A12	
GEO_CODE	A11	
IN_CITY_LIMIT	A1	

Statement Construct Maintenance

The statement construct enables you to indicate the collection of accounts and/or contracts whose financial details should be consolidated for a statement to a given person. The statement construct also contains routing information needed by the system to send the statement to the person in the preferred manner.

Lifecycle of a Statement Construct

The following diagram illustrates the lifecycle of a statement construct.



`Active` The statement construct will be in this status while a person is actively receiving statements for the accounts and contracts linked to this construct. A statement construct may transition from this status to `inactive`.

`Inactive` The statement construct will be in this status when a person no longer wishes to receive statements for the accounts and contracts linked to this construct. A statement construct may transition from this status back to `active`.

Copyright © 2024, Oracle and/or its affiliates.

Statement Construct - Main

Open this page using **Billing Management, Statement Construct**.

Description of Page

Statement Construct is a concatenation of summary information about this record. It is composed of the name of the statement person, the print description and the status.

Statement Construct ID is the unique system-generated identifier for this statement construct.

Person ID is the person who receives the future statements.

> **Note:** A person may be linked to many statement construct records.

Status indicates whether or not this person is currently receiving statements. The values are `Active` and `Inactive`.

> **Fastpath:** Refer to Lifecycle of a Statement Construct for more information.

Address Source - Used to indicate the source whose address should be used for sending the statements to the person.

Address ID - Used when you want to send the statements generated for the person to a particular address. If the address source is set to **Person**, you can only specify the address ID which is associated with the person. It can be either the person's mailing or seasonal address.

> **Note:**
>
> The **Search** (🔍) icon appears corresponding to this field. On clicking the **Search** icon, the **Address Search** window appears where you can search for the required address.
>
> If the address ID is not specified and address source is set to **Person**, the mailing address which is effective for the person at the time of statement generation is considered for statement routing.

Statement Cycle - Used to indicate when statements are produced for this statement construct. Refer to Statement Cycle for a description of how the production schedule is maintained.

Statement Route Type - Used to indicate how the statement is routed to the **Person** (e.g., via email, postal service, EDI, etc.). Refer to Statement Route Type **for more information** on setting up statement route types.

Number of Copies - Used to indicate how many copies of the statement the person wishes to receive.

Statement Format - Used to indicate if the person receives a `Detailed` or a `Summary` statement.

> **Note:** The values for this field are customizable using the Lookup table. The values need to match the formats supported by your statement print software. This field name is STM_FORMAT_FLG.

Print Description - Used to specify a brief description that can be printed on the statement.

The tree shows the statement construct's details. The nodes expand to show account and contract information for the construct details.

> **Note: Large statement constructs** The tree is hard to use if the statement construct has many details. In this situation, transfer to the adjacent tab page and use the filter to restrict the details that are shown.

Statement Construct Identifier - Used to specify the value for the statement construct.

Statement Construct - Details

This page is used to maintain the accounts and contracts where financial transactions should appear on statements produced using this statement construct. Open this page using **Billing Management, Statement Construct, Details**.

Copyright © 2024, Oracle and/or its affiliates.

Description of Page

Statement Construct is a concatenation of summary information about this record. It is composed of the name of the statement person, the print description and the status.

Statement Construct ID is the unique system-generated identifier for this statement construct.

If a statement construct has a large number of details, you can use the **Details Filter** to limit the details that appear in the grid. The following options are available:

- `All.` Use this option to view all details linked to the statement construct.
- `Person Name.` Use this option to restrict details linked to accounts whose main customer has a primary name that matches an input **Name**.
- `Statement Print Description.` Use this option to restrict details to those with a given **Description**.

Don't forget to click the search button after changing the filter.

The grid contains the accounts and/or contracts whose financial information contributes to this statement construct. Each record in the grid contains the following information:

- **Construct Detail Type** defines if the construct detail is for an entire `Account` or a single `Contract`. This field is protected if a statement detail was produced using this construct.
- **Contract/Account** identifies the account (for `Account` construct details) or the contract (for `Contract`) construct details. This field is protected if a statement detail was produced using this construct.
- **Statement Print Description** is available for use by the extract program to include on the printed statements to help the recipient identify the details.
- **Print Order** control where the financial transactions associated with the account / contract appear on the printed statement.
- **Start Date** is the date that this account / contract's financial transactions should begin appearing on statements.
- **End Date** is the date that this account / contract's financial transactions should stop appearing on statements.

Note: Only financial transactions linked to bills whose BILL DATE is within the construct detail's **Start Date** and **End Dates** will appear on statements.

- **Construct Detail ID** is a system generated unique identifier of the construct detail.

Statement Construct - Characteristics

This page is used to add, edit, or delete the characteristic(s) which is associated with the statement construct. Open this page using **Billing Management, Statement Construct, Characteristics**.

Description of Page

Statement Construct is a concatenation of summary information about this record. It is composed of the name of the statement person, the print description and the status.

Statement Construct ID - Used to search the unique system-generated identifier for this statement construct.

Note: To define the characteristic for a statement construct, you should have:

- Characteristic types defined in the application (where the characteristic entity is set to **Statement Construct**)

The Characteristics section contains the following fields in a grid:

Effective Date - Used to specify the date from when the characteristic is effective for the statement construct.

Characteristic Type - Used to indicate the characteristic type.

Note: The list includes only those characteristic types where the characteristic entity is set to **Statement Construct**.

Characteristic Value - Used to specify the value for the characteristic type.

Copyright © 2024, Oracle and/or its affiliates.

> **Note:** If you select a predefined characteristic type, the **Search** (🔍) icon appears corresponding to the **Characteristic Value** field. On clicking the **Search** icon, the **Predefined Characteristic Search** window appears. On specifying the value for a predefined characteristic type, the description of the characteristic value appears corresponding to the **Characteristic Value** field.

If you want to define more than one characteristic for the policy, click the **Add** (+) icon. However, if you want to remove a characteristic from the policy, click the **Delete** (🗑) icon corresponding to the characteristic.

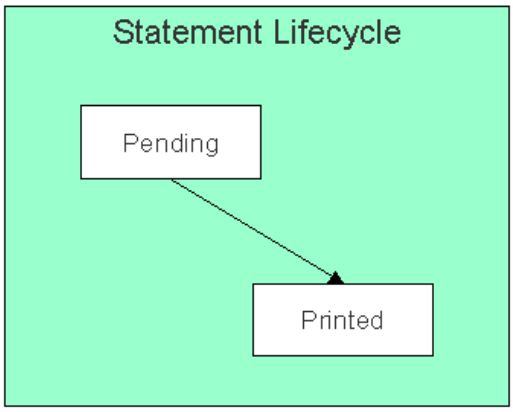
Statement Maintenance

Statements allow you to set up a Person to receive a consolidated report of the financial activity for one or more accounts and/or contracts. The topics in this section describe how to maintain statements.

> **Note: The system creates most statements behind-the-scenes.** Most statements are created by the system when it processes the statement cycle schedule. You should only have to access the statement pages to regenerate a statement or to add a statement real-time. For information about how the system creates statements, refer to The Statement Creation Background Process.

Lifecycle of a Statement

The following diagram illustrates the lifecycle of a statement.



`Pending` A statement is created in this state and will remain in this state until the statement details are extracted. When that occurs, it will transition to `printed`.

`Printed` A statement transitions to this state after the details of a statement have been extracted.

Statement - Main

Open this page using **Billing Management, Statement**.

> **Note: Correcting erroneous statements.** It's important to be aware that there are very few fields that are directly modifiable by a user. This is because a statement is a summary of financial transactions. If you need to change information on a statement, you must change the source information (e.g., change the statement's construct or complete additional bills) and then regenerate the statement. Refer to How To Regenerate A Statement for more information.

Description of Page

Statement is a concatenation of summary information about this record. It is composed of the name of the statement person, the statement construct print description, the create date of the statement and the statement status.

Copyright © 2024, Oracle and/or its affiliates.

Statement ID is the unique system-generated identifier for this statement.

Indicate the **Construct ID** for the statement construct whose construct details should be included in this statement.

Address is a display-only field that contains the address to which the statement will be sent.

The **Statement Status** indicates the current status of the statement. The values are `Pending`, `Printed`.

Fastpath: Refer to Lifecycle of a Statement for more information.

The **Create Date** indicates the date that this statement was created.

The **Batch Control** and **Batch Number** are the batch process and run in which the statement was (or will be) sent to the recipient.

Fastpath: Refer to Printing Statements and to Statement Route Type for more information.

The **Statement Action** area contains buttons that you use to generate, reprint or delete a statement.

- The **Generate** button enables you to create a statement on-line. Refer to On-line Statement Production for more information.
- The **Reprint** button will stamp the latest run number for the statement's batch control. This will enable the statement to be reprinted the next time the batch control process executes. Refer to How to Reprint A Specific Statement for more information.
- The **Delete** button will remove the statement and its details from the database. Refer to How to Regenerate a Specific Statement for more information.
- The **Display** button will display an online image of the statement when pressed. Refer to Online Statement Image for more information.

Note: You can only use the **Display** button if your system has been configured to display an on-line image; otherwise, a message indicating that the service is not available will appear. This option can only be configured by your technical staff. Refer to Technical Implementation Of Online Statement Image for more information.

The bottom portion of this page is dedicated to a tree that shows the statement details. The nodes expand to show every contract that will appear on the statement and the financial details that will be swept onto the statement for each contract. You may transfer to the details of any of the nodes by selecting that node. Context menus are also available for various nodes to facilitate easy navigation to other pages.

Note: Large statements. The tree will be hard to use if the statement has many contracts that contributed financial transactions to the statement. In this situation, transfer to the adjacent tab page and use the filter to restrict the contracts that are shown.

Statement - Details

This page is used to view the contracts that contributed financial transactions to the statement. Open this page using **Billing Management, Statement** and then navigate to the **Details** tab.

Description of Page

Statement is a concatenation of summary information about the statement. It is composed of the name of the statement person, the statement construct print description, the create date of the statement and the statement status.

Statement ID is this statement's unique system-generated identifier.

If a statement has a large number of contracts that contributed financial transactions to the statement, you can use the **Contract Filter** to limit the contracts that appear in the grid. The following options are available:

- `All.` Use this option to view all contracts that contributed financial transactions to the statement.
- `Person Name.` Use this option to only show contract linked to accounts whose main customer has a primary name that matches **Person Name**.

Copyright © 2024, Oracle and/or its affiliates.

- `Statement Print Description.` Use this option to only show contracts associated with a construct detail with a given **Statement Print Description**.

Don't forget to click the search button after changing the filter.

The grid contains contracts that contributed financial transactions to the statement. If a contract's financial transactions appeared on multiple bills, a separate line is displayed for each bill. You can press the go to button adjacent to **Bill Due Date** to be transferred to Financial Transactions On A Bill where the individual financial transactions associated with the associated contract and bill can be viewed. The following information is displayed in the grid:

- **Statement Print Description** is the description of the construct detail associated with the contract.
- **Bill's Due Date** is the date of the bill on which the contract's financial transactions appear. Press the adjacent go to button to be transferred to Financial Transactions On A Bill where the individual financial transactions can be viewed.
- **Contract Information** describes the contract that contributed financial transactions to the statement.
- **Account Information** describes the account associated with the contract whose details appear on the statement.
- **Construct Detail ID** is the unique identifier of the construct detail associated with the contract.
- **Statement Detail ID** is the unique identifier of the statement detail on which the contract's financial transactions appear.

Copyright © 2024, Oracle and/or its affiliates.

Chapter 8

Sales & Marketing

Topics:

- The Big Picture of Campaigns, Packages and Orders
- The Big Picture Of Package Eligibility Rules
- The Big Picture Of Campaign Eligibility Rules
- Designing Campaigns and Packages
- Other Useful Information
- Maintaining Orders
- Maintaining Campaigns
- Maintaining Packages

The sales and marketing functionality satisfies many diverse requirements. For example, you can use this functionality to:

- Enroll new customers using a single transaction.
- Sell new products to existing customers.
- Update person and account information using a single transaction.
- Market your services to prospects from a marketing list (and measure the success of your efforts). If the customer responds to your sales efforts, the system will automatically setup the customer related contracts.
- Setup marketing surveys and record your customers' responses.
- Quickly create one-time charges.
- Setup proposals for prospective services (and then send a quotation to the customer for these services).
- And more...

The topics in this section provide more information about the sales and marketing functions.

The Big Picture of Campaigns, Packages and Orders

Three objects are at the heart of the sales and marketing functionality: **Campaigns**, **Packages**, and **Orders**.

- A **package** defines a basket of goods and services that can be offered to a customer or a prospect.
- A **campaign** is a structured effort to offer a given set of packages to existing customers or new prospects.
- An **order** is created for each customer / prospect to which a campaign is targeted.

The topics in this section provide background information about these objects.

An Overview of Campaigns, Packages and Orders

An order is used to define:

- Demographic information about a customer / prospect.
- Geographic information about the service address.
- The customer's response to eligibility-oriented questions. For example, you can pose questions like: Who is your current service provider? / Would you like to pay automatically? / What is your date of birth?

After the above information is defined, the order transaction presents packages that may be offered to the customer. A package controls the various types of contracts that will be created if the customer selects the package.

Note: The customer must be eligible for a package. When you setup a package, you define its eligibility criteria. For example, you can setup a package that is only applicable to customers in Toronto who pay automatically and who commit to a one-year service contract.

If the customer elects to take a package, the order transaction sets up / updates the "V" (along with all of the ancillary things that happen when service is initiated). Please note that in addition to setting up the "V", it's also possible to populate / update other information when a package is selected. For example, you could have the system setup the customer's automatic payment options.

Note: An order can be completed without creating contracts. It is possible to use the order transaction to simply create / update persons and accounts. Refer to Marketing Surveys and Setting up a New Customer Prior To Using Start/Stop for more information.

All orders must reference a campaign. An order's campaign defines:

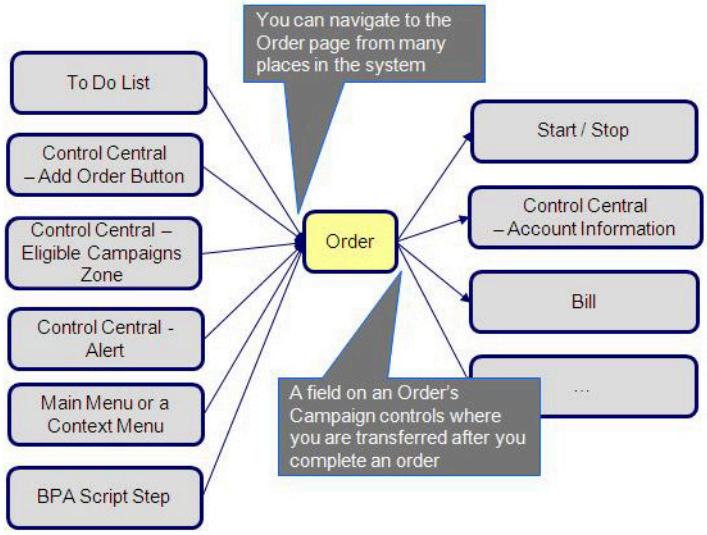
- The eligibility-oriented questions that are posed to the customer when an order is taken. For example, the questions indicated above - Who is your current service provider? / Would you like to pay automatically? / What is your date of birth? - are all defined on the order's campaign.
- The superset of packages that can be offered to a customer whose order references this campaign. An order's campaign defines the types of packages that may be selected.
- In addition to the above, campaigns also control high-level *eligibility rules* and business process flows.

Note: Bottom line. Campaigns allow you to define a group of **packages** that a customer is eligible to purchase. Every time a campaign is targeted at a customer, an **Order** is created. The information entered on the order qualifies the customer for one of the campaign's packages. If a customer elects to take a package, the "V" is setup / updated along with all of the ancillary things that happen when service is initiated, workflow processes are initiated, etc.

Supported Business Processes

Before providing examples of specific business processes, we'd like to highlight that you have control over some aspects of the user-interface flow (i.e., the screen navigation is not hard-coded). The following illustration highlights the potential user-interface flow:

Copyright © 2024, Oracle and/or its affiliates.

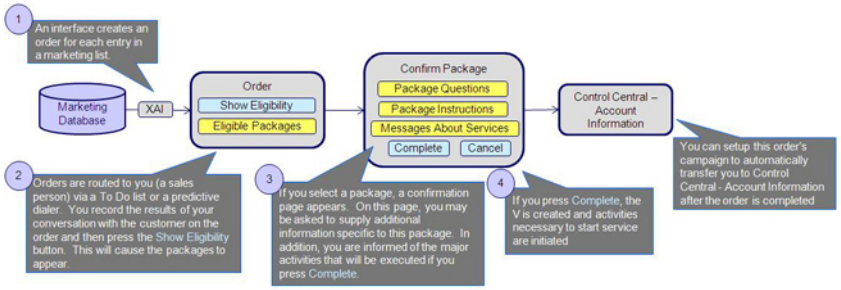


Note: Bottom line. You can navigate to the order transaction from many places in the system. After you complete an order, you are transferred to an appropriate transaction. The specific transaction is controlled by you when you setup the order's campaign.

The topics that follow illustrate specific business process flows that are possible in your implementation of the sales and marketing functionality.

Marketing to Prospects from a Marketing List

The following business process flow illustrates how the sales and marketing functionality would be used to market to prospects uploaded from a marketing database.



The following points describe the steps in this business process:

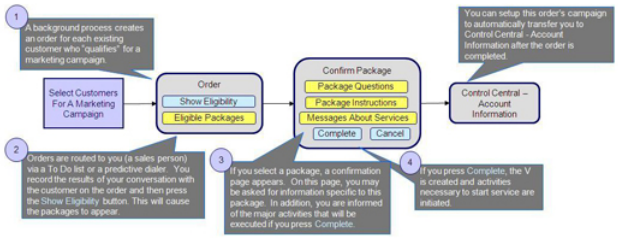
- An upload interface creates an order for each entry in a marketing database.
- Each order is routed to a salesperson via a To Do entry (or via a predictive dialer if your company supports this type of technology).
- The first step in the salesperson's interaction with the prospect will be to confirm the information uploaded from the marketing list (e.g., is your Bob Smith?, is your address 15 Main St?, etc.). Any changes are made to the information saved on the order.
- The salesperson's next step involves posing eligibility-oriented questions related to the campaign. Examples of questions include:
-
  - **What is your date of birth?** This type of question might be asked if you have packages that are only marketed to senior citizens. Potential answers would be a valid date that's in the past.
  - **Would you like to pay automatically?** Potential answers are: Yes, No.
  - **Would you like to pay using a levelized payment plan?** Potential answers are: Yes, No.
  - Etc.

Copyright © 2024, Oracle and/or its affiliates.

- The gathered information is saved on the order and then the system displays the packages that can be offered to the customer. The available packages may be restricted based on information on the order.
- If the customer is interested in a package, the salesperson selects it. Prior to the order being completed, the package confirmation page is shown. On this page are additional instructions about the package and a description of how the system will be updated if the package is selected (e.g., the types of contracts will be displayed). In addition, the package may require additional information before it can be chosen. For example, the package may only be selectable if the customer agrees to pay automatically. In this situation, the package confirmation page will require the salesperson to enter the customer's bank and bank account.
- After entering all package-specific information, the order is completed. When the order is completed, the "V" is setup and all work necessary to satisfy the order is initiated (e.g., workflow processes are initiated, etc.).

Marketing to Pre-selected Customers

The following business process flow illustrates how the sales and marketing functionality would be used to market additional services to existing customers selected by a background process.



The following points describe the steps in this business process:

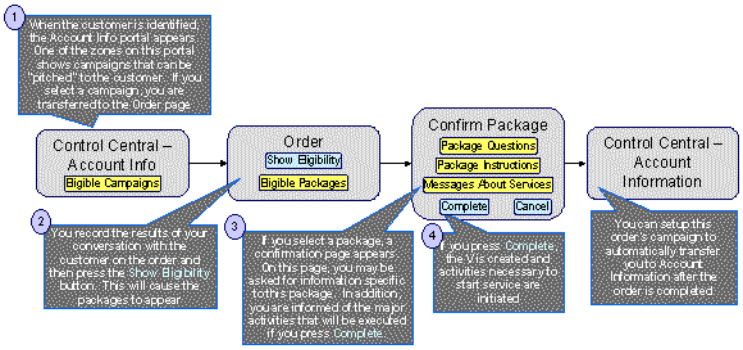
- A background process creates an order for each customer to be targeted under a marketing campaign.

> **Note:** The base package does not have an example of such a background process. Please speak to your support team if you require assistance in developing this type of process.

- Sales people would then process the order as described above.

Real-time Marketing of Additional Services to a Customer

The following business process flow illustrates how the sales and marketing functionality would be used to market additional services to existing customers when they call in.



The following points describe the steps in this business process:

- When you find the customer's record on Control Central - Main, you are automatically transferred to the Account Information portal. One of the zones on this page contains the campaigns that can be offered to the customer.

> **Note:** The "potential campaigns" zone only appears if the CSR has modified their preferences to display this zone.

- If you select a campaign from this zone, you are transferred to the Order transaction.

Copyright © 2024, Oracle and/or its affiliates.

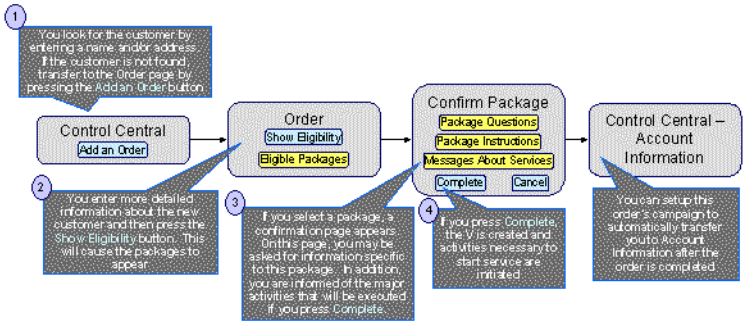
> **Note:** An order's campaign controls whether the Main or Questions & Misc Fields tab is initially displayed when the order transaction is invoked from this zone. For example, you might want to have the **Main** tab displayed if you want the order-taker to confirm the customer's demographic information before posing questions. Alternatively, you may want to skip this step and immediately present the order-taker with the campaign's questions. Keep in mind that the user can always display the other tab regardless of what is initially displayed.

- You then process the order as described above.

> **Note: An easy way to create one-time charges.** The above business process illustrates how the selection of a package will create one or more new contracts for a customer. If your organization levies one-time charges (e.g., tree trimming charges, damage assessments, etc.), you can also use the order transaction to create both a "one-time charge" contract and create the related billable charge when the order is completed. To do this, setup a campaign with a separate package for each possible one-time charge. If an order is created for this type of campaign and the relevant package is selected, the system will setup the new billable charge contract and link the billable charge to it. Refer to An Easier Way To Create One Time Charges for an example of such a campaign.

Real-time Marketing of Services to a Prospect

The following business process flow illustrates how the sales and marketing functionality would be used to market services when a new customer calls.



The following points describe the steps in this business process:

- You look for the customer using Control Central - Main. Because the customer is new, you push the Add Order button on Control Central to transfer to the Order transaction.
- You then process the order as described above.

In order to take advantage of the above business process, it's important that you understand the following points:

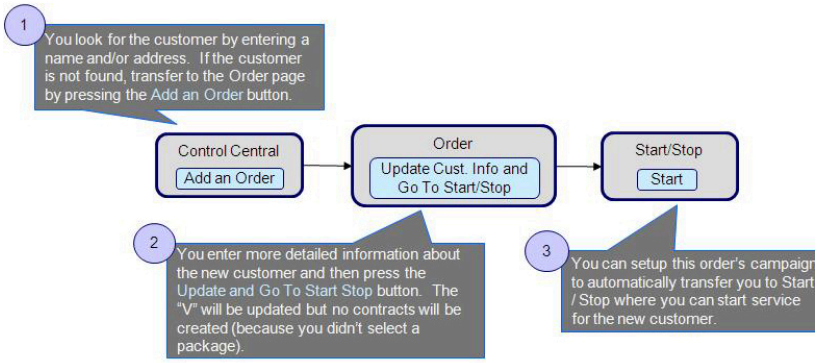
- When the order page is opened for a new customer, the order's campaign is defaulted from the installation record. The default campaign on the installation record is "generic" because we don't know anything about the customer (they are new) and therefore we cannot default a campaign targeted to their customer class or geography.
- This default campaign on the installation record must be constructed using one of the following approaches:
- - The default campaign has a package for every potential type of customer who can call up. While this is possible, it's probably not realistic as the data setup burden required to define every type of package could be onerous. However, if you have a limited number of contracts, this approach is feasible.
  - The default campaign could just be a "dummy" campaign that *has no packages*. The "real" campaigns will only be shown after basic information about the customer has been recorded on the order and the "show eligibility" button is pressed. These campaigns appear in the order's eligibility tree. The selection of a campaign in this tree will cause the order's campaign to be switched. Note, because the "dummy" campaign has no packages, no packages will appear in the order's eligibility tree when the dummy campaign is referenced on an order.
- The order is then processed as described above.

Setting up a New Customer Prior To Using Start/Stop

The following business process flow illustrates how the sales and marketing functionality would be used to create a new customer prior to using Start / Stop to start service.

Copyright © 2024, Oracle and/or its affiliates.

Note: The Order transaction is used to create a new person and account rather than the Person transaction. The order transaction may be a better choice because it lets you define many attributes on person and account that are not accessible on the person page. For example, the order transaction lets you define the customer's automatic payment option, bill after date, override due date, etc. In addition, on an order's campaign, you can define default values of many field values (both real fields and characteristics). These default values are presented on the order transaction and may be overridden by the user at will.



The following points describe the steps in this business process:

- You look for the customer using Control Central - Main. Because the customer is new, you push the "Add Order" button on Control Central to transfer to the Order transaction.
- You record additional demographic and geographic information about the new customer and confirm all values defaulted from the order's campaign. After all required information is entered, you press the Update Cust. Info and Go To Start / Stop button to create the new person and account and transfer to Start / Stop to identify the new services desired by the customer.

Note: The button's existence and label are dynamic. In the above example, we indicated that there is a button on the order page called **Update Cust. Info and Go To Start / Stop**. This button's presence on the page and its label are controlled by fields on the campaign. When pressed, the "V" is updated with information on the order and the user is transferred to the "post completion" page (which is also defined on the campaign). You would only enable this button on campaigns whose orders can be completed without selecting a package. In the above example, the campaign doesn't have any packages (because service is intended to be started using Start / Stop). In Examples of Campaigns and Packages we show other examples of how these fields would be setup for different types of campaigns.

In order to take advantage of the above business process, it's important that you understand the following points:

- When the order page is opened as a result of pressing the Add Order button on Control Central - Main, the order's campaign is defaulted from the installation record. The default campaign on the installation record is "generic" because nothing is known about the new customer and therefore it is impossible to default a campaign targeted to their customer class or geography.
- The campaign on the installation record should be setup to contain default information for the "average" customer.
- When a user takes an order for a new customer, they'll change the default information when appropriate. For example, if an industrial customer were being added, they'd change the customer class to "industrial" and then ask to see the campaigns such a customer is eligible to choose (these alternate campaigns are displayed in the order's eligibility tree). For example, if you've setup campaigns with eligibility rules that are targeted at industrial customers and the customer class on the order has been changed to "industrial", the user will see these campaigns in the eligibility tree. The user can then change the campaign's order by simply clicking on the desired campaign.

Note: The above example illustrates how an order can be completed without causing the creation of contracts. Refer to Marketing Surveys and Prelude to Start Stop for examples of such campaigns.

Copyright © 2024, Oracle and/or its affiliates.

Examples of Campaigns and Packages

The topics that follow provide examples of campaigns and packages that support several different scenarios. Your organization's campaigns and packages may differ markedly from these examples. Factors that will influence your sales and marketing objects include:

- Whether the type of goods and services marketed to your customers / prospects differs depending on some trait of the customer or the region in which they are located. For example,
- - Your company may have very straightforward packages that only differ based on the type of customer. If this describes your company, the setup process will be simple.
  - Alternatively, your company might offer different packages to particular customers based on whether they pay automatically, whether the customer is a senior citizen, whether they purchase multiple services from you, etc. If this describes your company, then you'll have to define the various eligibility criteria and the respective packages applicable to customers who satisfy each criterion.
- Whether your company offers many different types of goods and services.
- Whether your rates have charges / discounts that are only applied to some customers. For example, you might have discounts that are applied if the customer is a senior citizen or an employee.
- Which of the Supported Business Processes you implement.

Use the information that follows to form an intuitive understanding of campaigns and packages. After attaining this understanding, you'll be ready to design your own campaigns and packages.

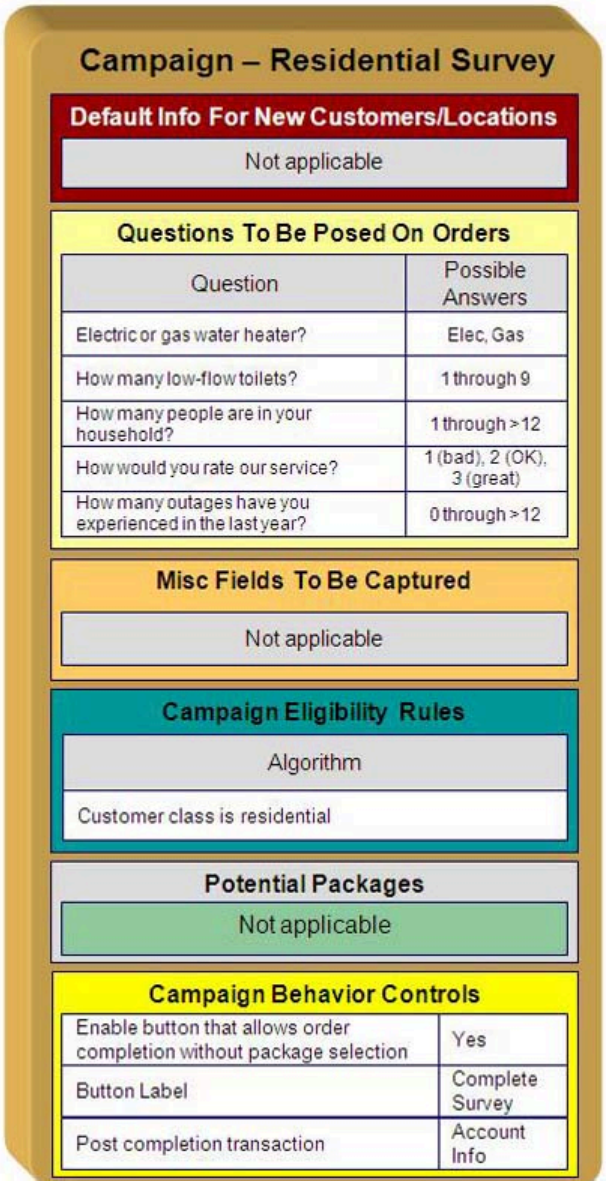
Campaigns Without Packages

An order that references a campaign without packages will never create contracts. The following topics describe scenarios when such a campaign would be used.

Marketing Surveys

Some organizations conduct "marketing surveys" to determine general trends amongst their customer base. The following diagram illustrates how such a campaign might look:

Copyright © 2024, Oracle and/or its affiliates.



Note the following about this campaign:

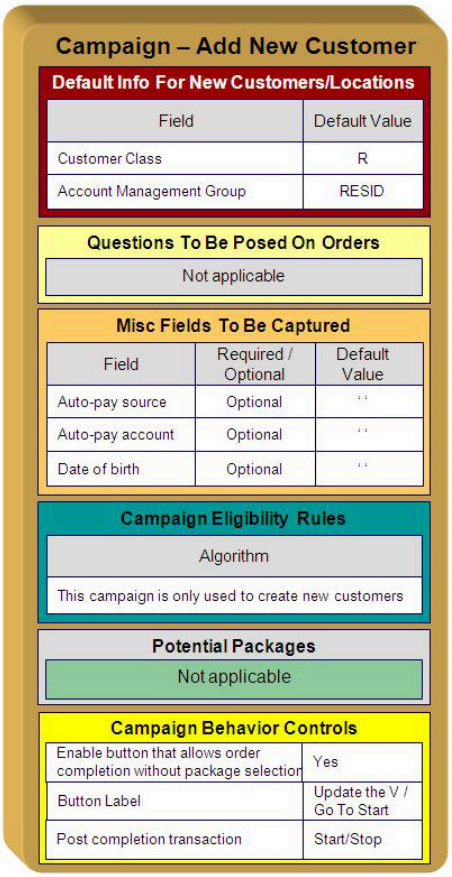
- If the campaign is used to establish *new* customers, you can define information to default onto orders created for new customers. Because marketing surveys are targeted at *existing* customers, you don't need new customer default information on the campaign.
- You can define questions to be posed to customers when an order is taken when you setup a campaign. For a "marketing survey" campaign, you'd define each of the survey questions and the permissible answers. When a customer is surveyed, an order is created and their answers are saved on the order.
- On a campaign, you can define additional fields to be updated on persons and accounts when orders associated with the campaign are completed. For example, you could indicate that the account's override due date and automatic payment options should be confirmed when an order is taken. Because marketing surveys are typically only asking questions, you wouldn't need to define additional person information fields to be confirmed. However, you can if you want.
- All campaigns need at least one eligibility rule. These rules also control whether the campaign appears on an order's eligibility tree (campaigns that appear in this tree are alternate campaigns that may be used on the order). Our example has assumed that this is a particular customer survey and therefore it only needs a single eligibility rule. Refer to *The Big Picture Of Campaign Eligibility Rules* for more information.

Copyright © 2024, Oracle and/or its affiliates.

- Because this campaign is used purely to manage a marketing survey, it doesn't need any packages. Packages are only necessary if contracts are created when an order is completed.
- Because this campaign doesn't have packages, we have indicated that the button that allows orders to be completed without selecting a package is enabled. If you don't enable this button, a user won't be able to complete a marketing survey (and it is only during order completion when the "V" is updated with the information on the order). Notice that we have assigned the button a label of Complete Survey.

Prelude to Start/Stop

Consider the example illustrated above under Setting up a New Customer Prior To Start / Stop. In this example, the order transaction is used to create a new person and account; no contracts are created when the order is completed. Rather, after the order is completed, the user is automatically transferred to Start / Stop where they can create contracts. The following diagram illustrates how such a campaign might look:



Note the following about this campaign:

- Because this campaign is used to establish *new* customers, you should define new customer default information. The order taker can override these default values.
- If we assume you don't ask marketing / eligibility-oriented questions when you take-on a new customer then this campaign would not require any questions.
- On a campaign, you can define additional fields to be updated on the related person and account when orders associated with the campaign are completed. For example, you could indicate that the account's override due date and automatic payment options should be confirmed when an order is taken. In our example, we are assuming that we want to define the customer's automatic payment options and his/her date of birth.
- As described under Marketing Surveys, a campaign's eligibility rule(s) control whether the campaign appears in the "eligible campaigns" and on an order's eligibility tree. This campaign is odd in that we don't want it to ever appear as an "eligible campaign" because it's only used for NEW customers. In order to make this work you must:
- • Define an eligibility algorithm on this campaign that prevents it from appearing as an "eligible campaigns". Refer to A Campaign That Is Never Eligible for more information.

Copyright © 2024, Oracle and/or its affiliates.

  - Update the installation record to indicate that this campaign is the one that should be used on orders created. Refer to Setting up a New Customer Prior To Starting Service for more information about the flow of this business process.
- Because this campaign is used purely to add accounts and persons, it doesn't need any packages. Packages are only necessary if contracts are created when an order is completed.
- Because this campaign doesn't have packages, we have indicated that the button that allows an order to be completed without selecting a package is enabled. If you don't enable this button, a user won't be able to complete the order (and it is only during order completion when the "V" is updated with the information on the order). Notice that we have assigned the button a label of Update The V and Go To Start.
- This campaign should be setup to transfer the user to Start / Stop after the order is completed.

Campaigns With Simple Packages

As explained under Campaigns Without Packages, there are several situations that might necessitate a campaign without packages. In this section, we describe simple scenarios of campaigns with packages.

An Easier Way To Create One Time Charges

We use the term "one-time charge" to describe ad hoc charges, e.g., tree trimming charges, damage assessments, etc. The following points describe how one-time charges are levied if you don't use the order transaction:

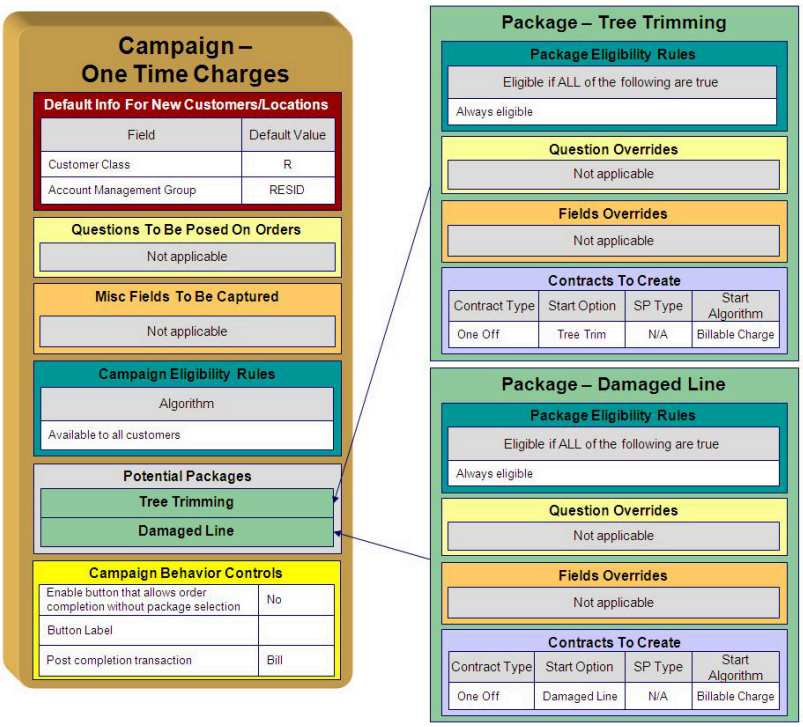
- Determine if the customer to be invoiced already has an appropriate "one-time charge" contract. If one doesn't exist, you must create one using Start/Stop Service.
- Next, a billable charge is added to the "one-time charge" contract (using Billable Charge Maintenance). The billable charge contains the invoice lines and amounts. The system will produce the bill for this charge when the account's bill cycle next executes.

Rather than going through the above multi-step process, you can also use the order transaction to create both a "one-time charge" contract AND create the billable charge. In addition, if the customer receiving the charge is new, the order transaction will also create a new person and account at the same time.

To take advantage of this facility, you must set up a campaign that has a separate package for each type of one-time charge as illustrated below. When an order is taken for such a campaign, the user simply selects the package that corresponds with the desired one-time charge. The system will then create the necessary customer and financial information when the package is confirmed.

Note: Billing of one-time charges. The order transaction will not cause a bill to be generated for the customer. Rather, the next time the account is billed, the system will see that there's an unbilled billable charge and create a bill segment for it. This bill segment will appear on the customer's regular bill along with their other charges. If you'd prefer to create the bill for the one-time charges immediately, you could setup the campaign to automatically transfer the user to the bill transaction where they can create an ad hoc bill. Note, if you do this, you'll need to create a new plug-in algorithm on the billable charge contract type to automatically activate the contract when it is created (otherwise the user would have to manually activate the billable charge contract before the bill can be generated).

Copyright © 2024, Oracle and/or its affiliates.



Note the following about this campaign:

- One-time charges can be levied against both new and existing customers. This means that orders associated with this campaign may cause new customers to be created. Therefore, we have specified information to default onto orders for new customers on the campaign. Note that we defined "financial service-oriented" default values as the majority of one-time charges are levied against particular customers.
- If we assume you don't ask marketing / eligibility-oriented questions when you levy a one-time charge then this campaign would not require any questions.
- We have assumed that you are not capturing miscellaneous fields when you levy a one-time charge.
- As described under Marketing Surveys, all campaigns need at least one eligibility rule. We have assumed that this campaign's packages can be selected by all types of customers and have therefore specified an eligibility algorithm that will be true for all customers.
- Because this campaign has packages, we have indicated that the button that allows an order to be completed without selecting a package is disabled. This means a user must choose one of the packages to complete the order.
- You might want to setup this campaign to automatically transfer the user to the bill transaction after the order is completed. See the note above the illustration for more information.
- We have assumed that this campaign can be used to levy two kinds of one-time charges - a tree trimming charge and a damaged line charge - and therefore this campaign requires two packages. Note the following about each package:
-
  - A package's eligibility rules control whether the package can be selected when an order is taken. With more sophisticated campaigns, you may have packages that are only applicable to certain types of customers. In our example, we've assumed that these types of one-time charges can be levied against all customers and have therefore used "universal" eligibility criteria (i.e., there are no eligibility restrictions). Refer to A Package That Is Always Eligible for more information.
  - On a package, it's possible to define additional questions to be posed if a package is selected. In our example, there are no additional questions to be posed for the one-time charge packages.
  - On a package, it's possible to define additional fields to be captured if a package is selected. In our example, there are no additional fields to be captured.
  - On a package, you define the types of contracts to be created if a package is selected. In our example, we only need a single contract created / updated if the package is selected and therefore there's just one entry in **Contract's To Create**. Please note the following about the information used to create this contract:
- The **Contract Type** is one that's used for a billable charge contract.

Copyright © 2024, Oracle and/or its affiliates.

- We've indicated a **Start Algorithm** of `Billable Charge`. Refer to STRM-AS for an example of such an algorithm (type).

The Big Picture Of Package Eligibility Rules

The packages described under Examples of Campaigns and Packages have a variety of eligibility rules. Designing these rules can be easy or time-consuming; it all depends on the complexity of your business rules. We'll walk you through several examples to help you form an intuitive understanding of how to setup eligibility rules. Once you've acquired this intuition you'll be ready to design the eligibility rules for your own packages.

> **Note:** Don't confuse *package* eligibility rules with those associated with a *campaign*. Package eligibility rules control whether a customer is allowed to choose a package on the order transaction. Campaign eligibility rules control whether a campaign is highlighted on the "eligible campaigns" content zone and on an order's eligibility tree.

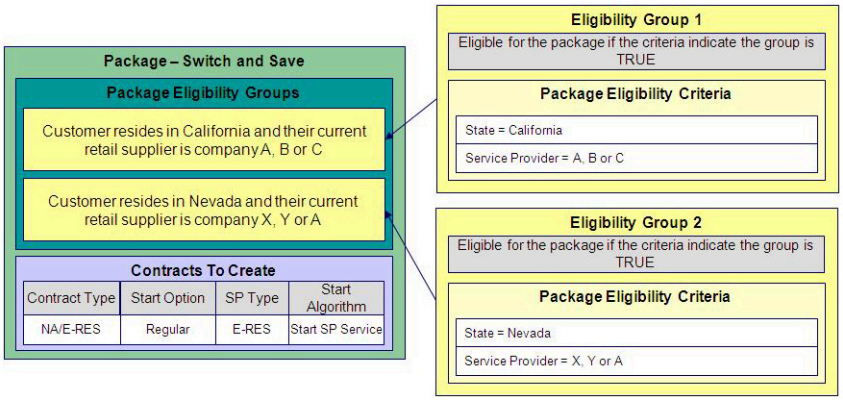
Criteria Groups versus Eligibility Criteria

Before we provide concrete examples of eligibility criteria, we need to explain two concepts: Criteria Groups and Eligibility Criteria. A package's criteria groups control whether a customer is eligible to choose a package. At a high level, it works like this:

- A criteria group has one or more eligibility criteria. A group's criteria control whether the group is considered TRUE or FALSE for a given customer.
- When you create a group, you define what should happen if the group is TRUE or FALSE. You have the following choices:
- - The customer is eligible to choose the package
  - The customer is not eligible to choose the package
  - The next group should be checked.

We'll use the following example to help illustrate these points. Assume a package exists that can only be selected if:

- The customer resides in California and their current retail supplier is company A, B or C
- OR, the customer resides in Nevada and their current retail supplier is company X, Y or A



This package requires two eligibility groups because it has two distinct conditions:

- IF (STATE = CALIFORNIA AND (SERVICE PROVIDER IN (A, B, C))
- IF (STATE = NEVADA AND (SERVICE PROVIDER IN (X, Y, A))

If either condition is true, then the customer is eligible for the package.

The following table illustrates how to create multiple criteria groups in an eligibility group:

Copyright © 2024, Oracle and/or its affiliates.

Group No.	Group Description	If Group is TRUE	If Group is FALSE
1	Customer resides in California and their current company is A, B or C	`Eligible`	`Check next group`
2	Customer resides in Nevada and their current company is X, Y or A	`Eligible`	`Ineligible`

The following table illustrates the criteria defined for the first criteria group:

Group 1: Customer resides in California and their current company is A, B or C				
Seq	**Logical Criteria**	**If Eligibility Criteria is TRUE**	**If Eligibility Criteria is FALSE**	**If Insufficient Data**
10	State = California	`Check next condition`	`Group is false`	`Group is false`
20	Financial service in (A, B, C)	`Group is true`	`Group is false`	`Group is false`

The following table illustrates the criteria defined for the second criteria group:

Group 2: Customer resides in Nevada and their current company is X, Y or A				
Seq	**Logical Criteria**	**If Eligibility Criteria is TRUE**	**If Eligibility Criteria is FALSE**	**If Insufficient Data**
10	State = Nevada	`Check next condition`	`Group is false`	`Group is false`
20	Financial service in (X, Y, A)	`Group is true`	`Group is false`	`Group is false`

The next section describes how you'd setup the specific logical criteria in each of the groups.

Defining Logical Criteria

When you setup an eligibility criterion, you must define two things:

- The field to be compared
- The comparison method

You have the following choices in respect of identifying the ***field to be compared*** :

- You can choose one of the miscellaneous fields on the order (keep in mind that miscellaneous fields also hold the answers to any questions posed on the order).
- You can execute an algorithm to retrieve a field value from someplace else in the system. This is a very powerful feature, but it's not terribly intuitive. We'll present a few examples later in this section to illustrate the power of this approach.

You have the following choices in respect of identifying the ***comparison method*** :

- You can choose an operator (e.g., >, <, =, BETWEEN, IN, etc.) and a comparison value.
- You can execute an algorithm whose job it will be to perform the comparison (and return TRUE, FALSE or INSUFFICIENT DATA). This is also a very powerful feature, but it's not terribly intuitive. We'll present a few examples later in this section to illustrate the power of this approach.

The Examples Of Package Eligibility Rules provide numerous examples to help you understand this design.

Copyright © 2024, Oracle and/or its affiliates.

Examples Of Package Eligibility Rules

The topics in this section provide examples about how to setup package eligibility rules.

A Package That Is Always Eligible

If a package is always eligible, it still needs an eligibility group. Good examples of such packages are shown under An Easier Way To Create One Time Charges.

The following table illustrates how to define the eligibility information in an eligibility group for a package that is always eligible:

Group No.	Group Description	If Group is TRUE	If Group is FALSE
1	Always eligible	`Eligible`	`Eligible`

The following table illustrates how to define a criteria in an eligibility group which is always eligible:

Group 1: Always eligible					
Seq	**Field to Compare**	**Comparison Method**	**If TRUE**	**If FALSE**	**If Insufficient Data**
10	Algorithm: retrieve order's account	Algorithm: Always true	`Group is true`	`Group is true`	`Group is true`

This group's criterion is odd in that it's not comparing anything. In order to setup this criterion, you have to use the following "trick":

- Field to Compare. You can pick any miscellaneous field on the campaign. If the campaign has no miscellaneous fields, you'll need to setup an algorithm to retrieve a field from the order (refer to PKEL-ENRFLD for an example of this type of algorithm). In the above example, we used an algorithm to retrieve the order's account.
- Comparison Method. We chose a comparison algorithm that always returns a value of `TRUE` (refer to PKCC-DEFAULT for an example of this type of algorithm).
- You'll notice that if a value of `TRUE` is returned, the `Group is true` (and we've setup the group to indicate a true group means the customer is eligible for the package).

We understand this is a little complicated, but this design provides the power necessary to support very complex criteria. The following sections provide more examples to help you form a better understanding of these concepts.

A Package With More Complex Operators

One of the eligibility groups described under Criteria Groups versus Eligibility Criteria has the following eligibility rules:

- Customer must live in California
- Service provider is A, B or C

The following table illustrates an eligibility group where the eligibility rules are defined:

Group No.	Group Description	If Group is TRUE	If Group is FALSE
1	Lives in California, current service provider is either retailer A, B or C	`Eligible`	`Ineligible`

The following table illustrates how to define a criteria in an eligibility group:

Copyright © 2024, Oracle and/or its affiliates.

Group 1: Lives in California, current service provider is either retailer A, B or C					
Seq	**Field to Compare**	**Comparison Method**	**If TRUE**	**If FALSE**	**If Insufficient Data**
10	Algorithm: retrieve order's state	= CA	`Check next condition`	`Group is false`	`Group is false`
20	Misc Field On Order	IN A, B, C	`Group is true`	`Group is false`	`Group is false`

This example is very simple. The only thing worth pointing out is that a number of operators can be used when you define a "comparison method". Refer to Package - Eligibility for the complete list of operators.

The Big Picture Of Campaign Eligibility Rules

In the previous section, we explained how to Design Eligibility Rules For A Package. In this section, we explain how to design eligibility rules for a *campaign*.

A campaign's eligibility rules control two things:

- Whether the campaign appears in the Eligible Campaigns content zone. This content zone suggests possible campaigns for the current customer displayed on control central. You might want to use this zone if you have different campaigns that are offered to different types of customers. Refer to Real-time Marketing of Additional Services to a Customer for an example of how this content zone can be used to "up sell" additional services.
- Whether the campaign appears on an order's eligibility tree. Campaigns that appear in this tree are alternate campaigns that may be used on the order. Refer to Real-time Marketing of Services to a Prospect for an example of how an order's eligibility tree can contain alternate campaigns for an order.

Designing these criteria can be easy or time-consuming; it all depends on the complexity of your business rules. We'll walk you through several examples to help you form an intuitive understanding of how to setup campaign eligibility rules. Once you've acquired this intuition you'll be ready to design the eligibility rules for your own packages.

Note: Don't confuse eligibility rules for a package with those associated with a campaign. Package eligibility rules control whether a customer is allowed to choose a package. Campaign eligibility rules control whether a campaign is highlighted on the Eligible Campaigns content zone and on an order's eligibility tree.

Campaign Eligibility Rules Are Not Strictly Enforced

While the system prevents ineligible packages from being selected for an order (based on the package's eligibility rules), the system does not strictly enforce a campaign's eligibility rules. In other words, a user can change the campaign on an order to any `active` campaign. This "laxness" is intentional; otherwise it could become impossible to change an order's campaign.

It might be more helpful to think of campaign eligibility rules as "highlight conditions". These "highlight conditions" simply control whether the campaign appears on an order's eligibility tree.

Campaign Eligibility Rules Are Defined Using Algorithms

You define a campaign's eligibility rules using one or more algorithms on the campaign. These algorithms are relatively straightforward as they simply return one of the following values:

- `Eligible`. Meaning the customer is eligible for the campaign.
- `Ineligible`. Meaning the customer is not eligible for the campaign

Copyright © 2024, Oracle and/or its affiliates.

- `Check Next Algorithm`. Meaning that the next algorithm should be checked. You'd only design an algorithm to return such a value if multiple algorithms are used to determine a customer's eligibility for a campaign.

> **Note: Bottom line.** A customer is eligible for a campaign if at least one of its eligibility algorithms returns a value of "eligible".

The Examples Of Campaign Eligibility Rules provide numerous examples to help you understand this design.

Examples Of Campaign Eligibility Rules

The topics in this section provide examples about how to setup campaign eligibility rules.

A Campaign That Is Always Eligible

If a campaign is always eligible, it would require a simple eligibility algorithm that returns an indication that it is always `eligible`. A good example of such a campaign is shown under An Easier Way To Create One Time Charges. Refer to CAEL-DEFAULT for an example of this type of eligibility algorithm.

A Campaign That Is Never Eligible

If a campaign is never eligible, it would require a simple eligibility algorithm that returns an indication that it is `ineligible`. A good example of such a campaign is shown under Prelude To Start / Stop. Refer to CAEL-DEFAULT for an example of this type of eligibility algorithm.

Designing Campaigns and Packages

Designing campaigns and packages can be simple or challenging. Factors that will influence the complexity of the design process include:

- Whether the type of goods and services marketed to your customers / prospects differs depending on some trait of the customer or the region in which they are located. For example,
- Whether your company offers many different types of goods and services.
- Whether your rates have charges / discounts that are only applied to some customers. For example, you might have discounts that are applied if the customer is a senior citizen or an employee.
- Which of the Supported Business Processes you implement.

The topics that follow provide a structured approach to design your campaigns and packages.

Determine The Place Of Start / Stop

Before you can design your campaigns and packages, you must determine if your organization will use start / stop to initiate new contracts or whether new service can be initiated using the order transaction. Refer to Order versus Start / Stop for information to help you make this decision.

If you decide to use start / stop to initiate service, we'd recommend creating the following types of campaigns:

- Create a campaign similar to that described under Prelude to Start / Stop.
- If you levy one-time charges, you may also want to create a campaign similar to that described under An Easier Way To Create One Time Charges.
- If your organization performs marketing surveys, you may also want to create a campaign similar to that described under Marketing Surveys.

If you decide to use the order transaction to initiate new service, follow the steps outlined in the following sections.

Copyright © 2024, Oracle and/or its affiliates.

Extract Algorithms From The Eligibility Rules

The following table lists the algorithms that are required for the above illustrated eligibility rules:

Algorithm Type	Algorithm	Parameter Values
Retrieve a field from the order for eligibility comparison (see PKEL-ENRFLD)	Retrieve Order's Customer Class	Field name: CUST_CL_CD
Retrieve a field from the order for eligibility comparison (see PKEL-ENRFLD)	Retrieve Order's Division	Field name: CIS_DIVISION
Time span comparison (see PKCC-TIMESPN)	Determine if birth date equates to that of a senior	Time span type: YEAR Operator: >= Comparison Value: 65

Extract Miscellaneous Fields From The Eligibility Rules

During this step, you extract miscellaneous fields that must be captured on the order in order to drive the eligibility rules.

The following table illustrates a miscellaneous field which is captured on the order to drive the eligibility rules:

Misc. Field	Where Stored	Why Is It Needed
Date of birth	Person characteristic	Numerous eligibility rules use this field to determine if a customer is a senior citizen

Note: Fields explicitly referenced on every order. You'll notice that we didn't list those fields referenced in the eligibility rules that are explicitly referenced on every order (e.g., customer class, state, division, etc.). This is because these types of fields come "for free".

Determine Additional Information That Is Required If A Package Is Selected

During this step, you determine if there's any information that needs to be created when a package is selected.

The following table lists the extra fields that are appended to the miscellaneous field table:

Misc. Field	Where Stored	Why Is It Needed
Date of birth	Person characteristic	Numerous eligibility rules use this field to determine if a customer is a senior citizen

Determine If There Are Additional Questions / Fields For Your Packages

In order to pay automatically several fields are required (Start Date, End Date, Autopay Source (i.e., bank / credit card), Account Number, Account Name, Expiration Date). These fields are not "explicitly referenced" on the order and therefore must be captured as miscellaneous fields.

The following table lists the fields which are included in the list of miscellaneous fields:

Copyright © 2024, Oracle and/or its affiliates.

Misc. Field	Where Stored	Why Is It Needed
Date of birth	Person characteristic	Numerous eligibility rules use this field to determine if a customer is a senior citizen
Auto Pay Start Date	Account / Autopay Start Date Column	Used if a customer pays automatically
Auto Pay End Date	Account / Autopay End Date Column	Used if a customer pays automatically
Auto Pay Source	Account / Autopay Source	Used if a customer pays automatically
Auto Pay Account Number	Account / Autopay Account ID	Used if a customer pays automatically
Auto Pay Account Name	Account / Autopay Account Name	Used if a customer pays automatically
Auto Pay Account Expiration Date	Account / Autopay Account Expiration Date	Used if a customer pays automatically

We're going to assume that we don't capture any additional marketing-oriented questions. If we did, we'd need additional fields to hold the response to each question.

Determine If Additional Instructions Should Be Shown

If a package is selected, you can present additional instructions to the user informing them of special tasks they should perform (or special instructions they should give to the customer). For example, you could instruct the user to inform customers who aren't senior citizens that they should call back if they become seniors as they may qualify for a better rate. These instructions can be defined on each package.

For the sake of brevity, we will assume no additional instructions are necessary when a package is selected.

Determine If There Are Campaign-Level Fields / Questions

Next, you determine if there are additional fields that need to be captured regardless of the package that's selected. For example, you may want to capture additional demographic or geographic information for marketing purposes. For instance, you ask the customer to rate the quality of service they have experienced (assuming they are a current customer). This additional question requires another entry in our list of miscellaneous fields.

The following table lists the additional fields that can be included in the list of miscellaneous fields:

Misc. Field	Where Stored	Why Is It Needed
Date of birth	Person characteristic	Numerous eligibility rules use this field to determine if a customer is a senior citizen
Auto Pay Start Date	Account / Autopay Start Date Column	Used if a customer pays automatically
Auto Pay End Date	Account / Autopay End Date Column	Used if a customer pays automatically
Auto Pay Source	Account / Autopay Source	Used if a customer pays automatically
Auto Pay Account Number	Account / Autopay Account ID	Used if a customer pays automatically
Auto Pay Account Name	Account / Autopay Account Name	Used if a customer pays automatically
Auto Pay Account Expiration Date	Account / Autopay Account Expiration Date	Used if a customer pays automatically

Copyright © 2024, Oracle and/or its affiliates.

Misc. Field	Where Stored	Why Is It Needed
Service Rating	Person characteristic	Campaign level marketing question

Determine The Properties Of Every Miscellaneous Field

At this point, you have amalgamated all miscellaneous fields required for the campaign and its packages. Now you have to define the following for each field:

- Prompt On Order. This is the prompt that will prefix the field when the order is taken. The following table provides examples.
- Campaign Applicability. This controls whether the field is `optional`, `required`, or `only applicable on package(s)`.
  - A value of `required` means the order taker must enter a valid value before packages are displayed for selection.
  - A value of `optional` means the order taker can leave the field blank prior to package selection. You can make an `optional` field `required` on individual packages. For example, you would probably make **Date of birth** required on the `Senior` package.
  - A value of `only applicable on packages` is used for fields that are only captured for specific packages. Fields of this type are suppressed prior to package selection. When you setup a package, you indicate if the field is applicable (if so, the user is only prompted for the field if the package is selected).
- Type Of Response. This controls where the field is stored when the order is completed and how it is validated. You have two options: `characteristic` or `column`.
  - If a field is a `Characteristic`, you must define which entity the characteristic value is to be stored on when the order is completed (`Person`, `Account` or `Order`). You must also define the **Characteristic Type** as this controls the characteristic value entered by the user.
  - If a field's value resides is a `column` (as opposed to a characteristic value), you must define the column reference. A field's column reference controls three things:
- How the field's value is retrieved. A field's value is retrieved when an existing person / account is referenced on an order.
- How the field's value is validated. A field's value is validated before package eligibility is determined and before the order is completed.
- How the field's value is posted (i.e., updated on the database). A field's value is posted when an order is completed.

Fastpath: Refer to *Extract Column References From Miscellaneous Fields* for more information about column references.

- Default Value On Order. This controls whether the order taker is supplied with a default value when an order is created. Note well: for `optional` and `only applicable on package` fields, you can specify a different default value on each package.

The following table lists the properties of the miscellaneous fields:

Misc. Field	Prompt On Order	Campaign Applicability	Type Of Response	Default Value On Order
Date of birth	What is your date of birth (used to qualify for senior discount)?	`Optional`	`Characteristic` **Entity:** `Person` **Char Type:** Date of birth	
Auto Pay Start Date	What day would you like to start automatic payment processing (YYYY-MM-DD)?	`Optional`	`Column` **Column Reference:** Auto Pay Start Date	

Copyright © 2024, Oracle and/or its affiliates.

Misc. Field	Prompt On Order	Campaign Applicability	Type Of Response	Default Value On Order
Auto Pay End Date	What day would you like to stop automatic payment processing (leave blank if not applicable)?	`Optional`	`Column` **Column Reference:** Auto Pay End Date	
Auto Pay Source	From what type of credit card / bank will the funds be debited?	`Optional`	`Column` **Column Reference:** Auto Pay Source	
Auto Pay Account Number	What is your credit card / bank account number?	`Optional`	`Column` **Column Reference:** Auto Pay Account Number	
Auto Pay Account Name	What is the name on your account?	`Optional`	`Column` **Column Reference:** Auto Pay Account Name	
Auto Pay Account Expiration Date	If paying with a credit card, what is the expiration date (MM-YYYY)?	`Optional`	`Column` **Column Reference:** Auto Pay Credit Card Expiration Date	
Service Rating	Please rate our service	`Optional`	`Characteristic` **Entity:** `Person` **Char Type:** Service rating	1 (excellent)

Extract Characteristic From The Miscellaneous Fields

The following table lists the characteristics where the data from the miscellaneous fields is stored:

Characteristic Type	Type	Valid Values	Entities On Which Characteristic Can Be Defined
Date of birth	`Ad hoc value`: use validation algorithm to confirm date is > 1-Jan-1900 and < 31-Feb-2002	Not applicable	`Person`, `Order`
Service Rating	`Predefined value`	0: Not applicable 1: Excellent 2: Very Good 3: Good 4: Poor	`Person, Order`

Copyright © 2024, Oracle and/or its affiliates.

Extract Column References From The Miscellaneous Fields

The following table lists the columns where the data from the miscellaneous fields is stored:

Column Reference	Retrieval Algorithm	Validation Algorithm	Post Algorithm	Foreign Key Reference
Auto Pay Start Date	Retrieve account's current automatic payment information	Validate automatic payment information	Post automatic payment information	
Auto Pay End Date	N/A - this field is handled by the Retrieval Algorithm on Auto Pay Start Date	N/A - this field is handled by the Validation Algorithm on Auto Pay Start Date	N/A - this field is handled by the Post Algorithm on Auto Pay Start Date	
Auto Pay Source	N/A - this field is handled by the Retrieval Algorithm on Auto Pay Start Date	N/A - this field is handled by the Validation Algorithm on Auto Pay Start Date	N/A - this field is handled by the Post Algorithm on Auto Pay Start Date	Auto pay source
Auto Pay Account Number	N/A - this field is handled by the Retrieval Algorithm on Auto Pay Start Date	N/A - this field is handled by the Validation Algorithm on Auto Pay Start Date	N/A - this field is handled by the Post Algorithm on Auto Pay Start Date	
Auto Pay Account Name	N/A - this field is handled by the Retrieval Algorithm on Auto Pay Start Date	N/A - this field is handled by the Validation Algorithm on Auto Pay Start Date	N/A - this field is handled by the Post Algorithm on Auto Pay Start Date	
Auto Pay Expiration Date	N/A - this field is handled by the Retrieval Algorithm on Auto Pay Start Date	N/A - this field is handled by the Validation Algorithm on Auto Pay Start Date	N/A - this field is handled by the Post Algorithm on Auto Pay Start Date	

There are three types of algorithms referenced in the above table:

- Retrieval algorithm. A column's retrieval algorithm retrieves the column's current value from the database. You'll notice that only one of our columns uses a retrieval algorithm. This is because we've used a sophisticated example of a group of columns that are retrieved at the same time. When you have such a group of columns, nominate one of the columns to do the retrieval work and specify the retrieval algorithm on it.
- Validation algorithm. A column's validation algorithm validates the column's value prior to the database being updated. You'll notice that only one of our columns uses a validation algorithm. This is because we've used a sophisticated example of a group of columns that are validated as a group. When you have such a group of columns, nominate one of the columns to do the validation work and specify the validation algorithm on it.
- Post algorithm. A column's post algorithm updates the column's value on the database when the order is completed. You'll notice that only one of our columns uses a post algorithm. This is because we've used a sophisticated example of a group of columns that are posted together. When you have such a group of columns, nominate one of the columns to do the posting work and specify the posting algorithm on it.

Notice that we have indicated that one of the columns has a "foreign key reference". You declare a "foreign key reference" if the valid values for a column are defined on another table in the system (this also affects the order user interface as a

Copyright © 2024, Oracle and/or its affiliates.

dropdown appears for columns that have foreign key references). In our example, Auto Pay Source is the only column whose valid values are defined on the automatic payment source control table.

Extract Algorithms From The Column References

The following table lists the algorithms that are required for the above illustrated column references:

Algorithm Type	Algorithm	Parameter Values
Retrieve automatic payment information from an account (see CRRT-AP)	Retrieve automatic payment information from an account	Param 1 = Auto Pay Start Date Column Reference Param 2 = Auto Pay End Date Column Reference Param 3 = Auto Pay Source Column Reference Param 4 = Auto Pay External Account Id Column Reference Param 5 = Auto Pay Expiration Date Column Reference Param 6 = Auto Pay Name Column Reference
Validate automatic payment information (see CRVL-AP)	Validate automatic payment information	Param 1 = Auto Pay Start Date Column Reference Param 2 = Auto Pay End Date Column Reference Param 3 = Auto Pay Source Column Reference Param 4 = Auto Pay External Account Id Column Reference Param 5 = Auto Pay Expiration Date Column Reference Param 6 = Auto Pay Name Column Reference
Post automatic payment information (see CRPS-AP)	Post automatic payment information	Param 1 = Auto Pay Start Date Column Reference Param 2 = Auto Pay End Date Column Reference Param 3 = Auto Pay Source Column Reference Param 4 = Auto Pay External Account Id Column Reference Param 5 = Auto Pay Expiration Date Column Reference Param 6 = Auto Pay Name Column Reference

Copyright © 2024, Oracle and/or its affiliates.

Determine Account Dependency For Each Column Reference

Later in the design process (see Determine How The Campaign Behaves), you define on the campaign whether an account is required, optional or not allowed on its orders. Your selection impacts how you setup column references. This is a little complicated so let's use an example:

The following table lists the various column references and their dependencies:

Column Reference	Question Dependency
Auto Pay Start Date	Must have account
Auto Pay End Date	Must have account
Auto Pay Source	Must have account
Auto Pay Account Number	Must have account
Auto Pay Account Name	Must have account
Auto Pay Expiration Date	Must have account

You'll notice that each column reference is account-oriented (this is because automatic payment fields are stored on the respective account).

Determine If Any Properties Are Overridden On Any Package

At this point, you have defined the properties of all miscellaneous fields at the campaign level. You now need to determine if any of your packages need to override any of these properties. In general, you can only override miscellaneous fields whose campaign applicability is `optional` or `only applicable on package(s)`.

CAUTION: If you override field properties on a campaign's packages, you will not be able to change the properties of the field at the campaign level.

Determine The Eligibility Rules For The Campaign

Note: Assumption. This section assumes you are comfortable with the information in The Big Picture Of Campaign Eligibility.

The easiest way to determine a campaign's eligibility is to examine each package's eligibility criteria and abstract common eligibility restrictions onto the campaign. It's important to remember that campaign-level eligibility is only used to highlight if the campaign MAY contain applicable packages; campaign-level eligibility is not strictly enforced.

The following table lists the two algorithms shipped with the product that are used to construct the campaign's eligibility criteria:

Algorithm Type	Algorithm	Parameter Values
Check if account's customer class is in a predefined list (see CAEL-CC)	If customer class is "financial service", check next algorithm	If customer's customer class is in list: 30 (check next) If customer's customer class is not in list: 20 (ineligible) List of customer classes: R
Check if account's division is in a predefined list (see CAEL-DIV)	If division is "California", customer is eligible	If customer's division is in list: 10 (eligible)

Copyright © 2024, Oracle and/or its affiliates.

Algorithm Type	Algorithm	Parameter Values
		If customer's division is not in list: 20 (ineligible) List of customer classes: CA

Determine How The Campaign Behaves

The final step before you're ready to setup the campaign involves defining a variety of behavioral functions as described below:

- You need to define how the campaign uses accounts. Specifically, on a campaign you define if an account is required / optional / not allowed on its orders. In addition, if an account is required or optional, you can control the new accounts. These controls prevent the unwanted proliferation of new accounts for campaigns that are targeted at existing accounts.
- Many fields on an order (e.g., phone type, customer class, division, account management group) can be defaulted on orders created for new customers. You will see these fields on Order - Main. Think about each such field and whether a default value is appropriate. Keep in mind that the order taker can override any of these default values.
- An order taker can indicate that they want to hold an order if they need to do further research before continuing with an order. A background process (referred to by the batch control ID `TD-ECBK`) will create a To Do entry for held orders as a reminder to get back in touch with the customer on a future date. You can define the default To Do role assigned to such To Do entries when you create a campaign. Keep in mind that this value can be overridden when the order taker holds an order.
- You can indicate on a campaign the transaction to which the user should be transferred when orders are completed. Refer to Supported Business Processes for examples of how you might want to use this field.

Setup Sequence

And now you're ready to set up your campaign. The following points describe the order in which this should be done:

- Setup any new column references required by the campaign's miscellaneous fields. You will have to leave the retrieve, validate and post algorithms blank until after the next step.
- Setup any new algorithms required by:
- - The packages' eligibility rules.
  - The campaign's eligibility rules.
  - The campaign's column references.
- Update the relevant column references with the appropriate retrieve, validate and post algorithms.
- Setup any new characteristic types required by the campaign's miscellaneous fields.
- Create the campaign using the information you designed above. Specifically:
- - You defined the question and miscellaneous fields under Determine The Properties Of Every Miscellaneous Field.
  - You defined the eligibility rules under Determine The Eligibility Rules For The Campaign.
  - You defined everything else under Determine How The Campaign Behaves.
  - We recommend leaving the campaign's status as `Inactive` until all packages are entered.
- Create the campaign's packages using the information you designed above. Specifically:
- - You defined the question and miscellaneous field overrides under Determine If Any Properties Are Overridden On A Package.
  - You defined the eligibility rules under *Determine Package Eligibility Rules*.
  - You defined the instructions under Determine If Additional Instructions Are Required.
- Change the campaign's status to `Active`.

Copyright © 2024, Oracle and/or its affiliates.

Note: The default campaign on the installation record. If the campaign you've setup is the "default" campaign used when a user presses the Add Order button on Control Central, make sure to specify it as the default campaign on the installation record. Refer to Setting up a New Customer Prior To Starting Service for more information.

Other Useful Information

The topics in this section contain miscellaneous information about the sales & marketing functions.

Order versus Start / Stop

There are two different ways to start service for a customer:

- As described under definitions, you can use the order transaction to start service.
- Alternatively, you can use start / stop.

The following table provides insight into the pros and cons of each method (after this table you will find our recommendations):

Issue	Order	Start / Stop
New customer setup	Information about the new customer can be entered on the order page. A person / account will be setup when the order is completed.	A new person and account must be setup prior to using start / stop.
Updates to existing person information	Changes to a customer's existing account and/or person info may be entered on the order page. The related person and/or account will be updated when the order is completed.	Start / stop cannot be used to update person or account information. Rather, you must transfer to the person or account page and make the necessary changes.
Incomplete orders	The order transaction allows you to record person information and hold it without starting service. The customer can call back in the future and you can continue where you left off.	You cannot leave a start half-finished - you either have to complete the start or cancel it.
Stop current service, start new service	The order transaction does not currently provide a mechanism for stopping service (it can only be used to start service).	Start / stop can be used to stop a customer's current service and start new service.
Changes after completion	An order cannot be modified after it is completed. Any changes (cancellations prior to starting service, changes of the start date, etc.) must be performed using the start / stop transaction.	The start / stop transaction allows you to cancel and make changes to pending start contracts.
Address changes	Address changes entered on the order transaction are applied when an order is completed (i.e., when a package is selected). They are not held awaiting contract activation.	When you enter an address change on the start / stop transaction, this address changed is saved until the first pending start or stop is activated. This means that the address change is effectively held until the start / stop is activated.

Based on the above, we'd recommend the following:

- We'd recommend using the order transaction to start service. If customers typically stop and start service in tandem, we'd recommend the following steps:
- • Stop the customers existing service using start / stop.

Copyright © 2024, Oracle and/or its affiliates.

- When you select the appropriate campaign, you'll be transferred to the order transaction where you can start the desired package for the customer.

Orders Cannot Be Changed After They Are Completed

The Lifecycle of an Order indicates that once an order is `complete`, it cannot become pending again. In addition to this rule, we'd like to stress that once an order is `complete`, you cannot make any changes to the order except to add new log entries.

To Do List Processing

In the base package, there are two To Do lists that may be affected by orders:

- Highlight pending orders. This To Do list contains orders that are in the `pending` state. Refer to Pending To Do Entries for more information about this To Do list.
- Highlight held orders. This To Do list contains orders that are in the `held` state. Refer to Held To Do Entries for more information about this To Do list. Refer to Order - Hold Dialog for more information about holding orders.

Hold and Cancel Reason Codes

In Lifecycle of an Order, we illustrated how a user can `hold` a `pending` order. Before a user can hold an order, they must supply an Order Hold Reason.

The same is true if a user wants to cancel a pending or held order (i.e., they must supply a Order Cancellation Reason).

These reason codes are purely for audit purposes.

Additional Things Can Happen When An Order Is Completed

An optional plug-in spot exists on customer class where you can introduce additional logic to be executed when an order is completed for an account that belongs to this customer class.

Another optional plug-in spot exists on campaign where you can introduce additional logic to be executed when an order is completed for the campaign.

New Account is Created if an Order is Completed

It's important to understand that an account is created when an order is completed. However, the system creates a person if you save an order so that an audit of the customer interaction exists in the system.

Account Usage is Controlled by the Campaign

When you set up a campaign, you define how the campaign's orders use accounts. Refer to the description of Account Usage under Campaign - Main for more information.

Maintaining Orders

The order transaction can be used to satisfy many diverse requirements. For example, you can use an order to:

- Enroll new customers using a single transaction.
- Sell new products to existing customers.

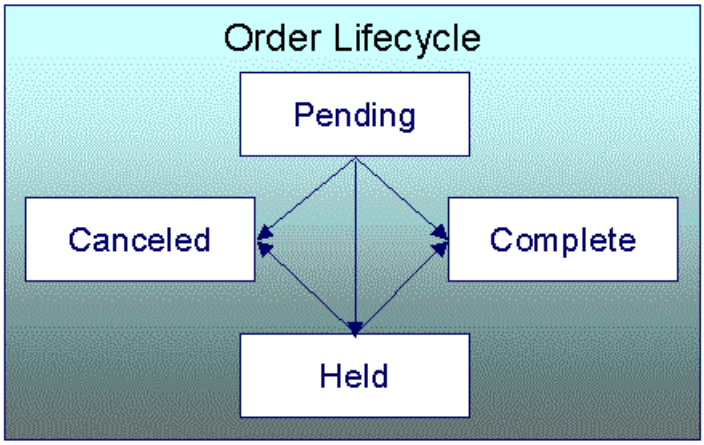
Copyright © 2024, Oracle and/or its affiliates.

- Update existing person and account information using a single transaction.
- Market your services to prospects uploaded from a marketing list (and measure the success of your efforts). If the customer responds to your sales efforts, the system will automatically setup the customer related contract.
- Setup marketing surveys and record your customers' responses.
- Quickly create one-time charges.
- Set up proposals for prospective services (and then send a quotation to the customer for these services).
- And more...

The topics in this section describe the order transaction. Refer to The Big Picture of Campaigns, Packages and Orders for general information about orders.

Lifecycle of an Order

The following diagram illustrates the lifecycle of an order.



`Pending` An order starts its life in the `Pending` state. An order remains in this state until you `Cancel`, `Complete` or `Hold` it.

`Held` You `Hold` an order when you want to save an order pending future information. For example, you would hold an order if you need to perform research on the perfect rate for the customer before completing the order. A `Held` order can be `Completed` or `Canceled`.

`Complete` When a customer selects a package of goods and services, the "V" is updated, contracts are initiated, and the order becomes `Complete`. No changes may be made to a `Complete` order.

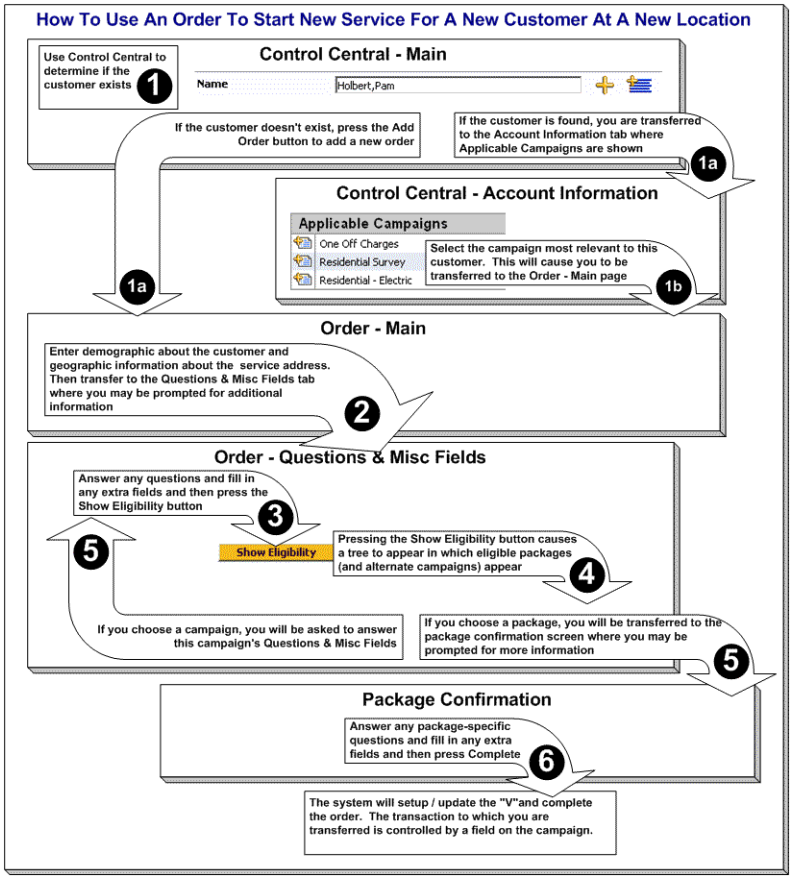
`Canceled` A `Pending` or `Held` order may be `Canceled`.

Order User Interface Flow

The diagram that appears below illustrates the anticipated navigation path used to process an order.

CAUTION: An order's campaign controls many aspects of the user interface flow. The illustration that appears below is a sample of what's possible; it may not mirror how your team has setup your company's campaigns.

Copyright © 2024, Oracle and/or its affiliates.



Order - Main

This page is used to define demographic and geographic information about the customer / prospect.

Fastpath: Refer to The Big Picture of Campaigns, Packages and Orders for more information about orders.

Open this page using **Sales & Marketing, Order**.

Fastpath: Refer to Order User Interface Flow for information about how to use this page.

Description of Page

Order contains a concatenation of the customer's name, campaign, package (if the order is completed), status and start date. **Order ID** is the system-assigned unique identifier of the order. These values only appear after the order is added to the database.

Status contains the state of the order. Refer to Lifecycle of an Order for information about the valid values and the state transition rules. Additional information may also be displayed based on the status:

- If the order's status is `Held`, the following information is also displayed: the hold reason, the name of the user who held the order, and the date and time when a reminder for the order is due to be created.
- If the order's status is `Canceled`, the following information is also displayed: the cancel reason, the name of the user who canceled the order, and the date and time when the order was canceled.
- If the order's status is `Complete`, the following information is also displayed: the name of the user who completed the order and the date and time when the order was completed.

Press the **Hold** button to hold a `Pending` order. Refer to Order - Hold Dialog for a description of the pop-up window used to hold an order.

Copyright © 2024, Oracle and/or its affiliates.

Press the **Cancel** button to cancel a `Pending` or `Held` order. Refer to Order - Cancel Dialog for a description of the pop-up window used to cancel an order.

If the **Campaign** has been setup to allow the completion of orders without selecting a package, a button appears to the right of **Cancel**. The label that appears in the button is defined on the **Campaign**. When pressed, the order is completed. When the order is completed,

- The "V" is setup / updated.
- All questions and miscellaneous fields (defined on the next tab) are posted to the appropriate object.
- Optional plug-ins are executed.
- You will be transferred to the order completion transaction that's defined on the **Campaign** (if any).

If the order references an existing person or account and the related person / account has changed after their information was snapshot on the order, the **Out of Sync** section appears. This section highlights which objects have changed since the order was last saved. In addition, the **Sync** button appears. When pressed, the order is updated with the current field values for these objects.

An order's **Campaign** controls many aspects of the order's behavior. For example, it controls:

- The type of information defaulted onto an order.
- The eligibility-oriented questions that are displayed on the next tab.
- The superset of packages that can be offered to a customer.
- In addition to the above, campaigns also control high-level *eligibility rules* and business process flows.

An order's **Campaign** is defaulted as follows:

- In all other situations, the order's campaign defaults from the Installation Record.

You may change an order's **Campaign** at will. Note that changing the campaign will result in the **Questions & Misc Fields** (the next tab) being refreshed with values associated with the new campaign. However, other values typically defaulted from the campaign (such as default customer class) do not get refreshed so that data you may have entered does not get overwritten.

Start Date defines the date on which service will start. This date becomes the start date on contracts created when this order is completed. If you need to change the start date on these contracts, transfer to Start/Stop (where all of the `pending start` contracts will appear).

The topics that follow describe the remaining sections on this page

Order - Main - Person Information

The **Person Information** section contains demographic information about the customer / prospect.

> **CAUTION:** This page contains a subset of person-oriented information. If you need to define other person-oriented information, transfer to Person - Main after the order is completed.

If this is a new customer, select `Create New Person` and enter demographic information in the fields that follow. Otherwise, select `Use Existing Person` and use the **Name** field or press the search button to find the customer. After you select an existing person, the following events takes place:

- The person's name appears beneath the person search (and the person context menu can now be used to drill down to person-oriented pages). Note, the person's name is formatted by a plug-in algorithm on the installation record. Refer to the base package's name format algorithm for an example. If you prefer different formatting logic, your system administrator should configure the system appropriately.
- If any of the Questions & Misc Fields (on the next tab) are used to capture person-oriented characteristics, the person's existing characteristic values are defaulted.
- If the selected person has at least one account, information about the account is displayed in the **Account Information** section (if the **Campaign** makes uses of accounts). In addition, if any of the Questions & Misc Fields (the next tab) are used to capture account-oriented characteristics, the account's existing characteristic values are populated in the respective rows on this tab.
- Information about the person is displayed in the fields that follow.

Copyright © 2024, Oracle and/or its affiliates.

Use **Person/Business** to indicate if the customer is a `Person` or a `Business`. This value controls how the person's primary name is validated. Note, for new customers, this value defaults based on the **Campaign**.

Names are used when you look for a customer based on their name. In addition, a person's primary name is the addressee on the person's bill unless overridden by the **Override Mailing Name** (defined on *Person - Contact Information*). To modify a name, simply move to a field and change its value. To add a new name, press + to insert a row, then fill in the information for each field. The following fields display:

- Use **Name Type** to indicate if the name is an `Alias`, `Alternate Representation`, `Doing Business As`, `Legal`, or `Primary` name. Note, for new persons, a value of `Primary` is defaulted.
- Use **Person Name** to define the person's name. Note well, the name is case sensitive.

Note: Alternate representations of a person's name. You would use an `Alternate Representation` for a person's name when you have an alternate ways to define the person's primary name. Alternate representations are typically used in countries that use multiple character sets (e.g., the `Primary` name is entered in Chinese, the `Alternate Representation` is entered in English). When a person has an alternate name, both the main and alternate names can be used to search for a person. The `Alternate Representation` **Name Type** only appears if you have enabled alternate addresses on the installation record. Refer to the description of the Alternate Representation field under Installation - Main for more information.

Note: Validation is performed by a plug-in. The validation that is applied to **Person Name** (e.g., a comma separating the last and first name - `Smith,Patricia`) is controlled by a plug-in algorithm on the installation record. Refer to the base package's name validation algorithm for an example. If you prefer different validation logic, your system administrator should configure the system appropriately.

Phone numbers are used to look for accounts and persons. For new customers, the phone type(s) default based on the **Campaign**. To modify a telephone number, simply move to a field and change its value. To add a new telephone number, press + to insert a row, then fill in the information for each field. The following fields display:

- **Phone Type** indicates the type of phone number, e.g., Home, Mobile, Business,...
- Use **Phone Number** to define the telephone number. Enter the telephone number in the format described by the **Phone Format**.

Note: Formatting is performed by a plug-in. The format that is applied to a **Phone Number** is controlled by the algorithm that is plugged in on the respective Phone Type. You can also have this plug-in convert input values into formatted values, e.g., this algorithm could transform all numbers into a formatted string. If you prefer a different format, your system administrator should configure this algorithm appropriately.

- Enter the **Extension**, if any, of the telephone number.

A person's **ID** has several uses:

- It is used when you look for a customer based on their ID.
- It is used to highlight potential duplicate persons.
- It displays a person's primary identification in the search results area to help a user identify the customer when multiple customers match the search criteria.

The person ID usage flag on the installation record indicates whether or not at least one id for a person is `required` or `optional`. The ID Type defaults from the installation record based on the **Person Type** (`Person` versus `Business`). The following fields are used to define a customer's primary ID.

- Indicate the type of identification in the drop down.
- Enter the identification number in the adjacent fields. Please note that if the ID number should be formatted (e.g., dashes in an American social security number), you do not have to enter the dashes. Rather, you can enter the information as a contiguous value and the system will format this for you. The format is shown in the adjacent **Identifier Format** column.

Copyright © 2024, Oracle and/or its affiliates.

> **Note: Formatting is performed by a plug-in.** The format that is applied to a person ID number(e.g., dashes in an American social security number) is controlled by the algorithm that is plugged in on the respective ID Type. If you prefer a different format, your system administrator should configure this algorithm appropriately.

> **Note: A person can have many forms of identification.** The order transaction only caters for a single form of ID (as this is the most typical situation). If you want to specify additional person ID's, you must transfer to Person - Main after the person is created (persons are created when the order is saved).

Specify the customer's **Email ID** (if any).

Define the **Language** in which the person prefers their bills and correspondence printed.

> **Note:** The person's language defaults from Installation Options - Person.

> **Fastpath:** Refer to Customer Language for more information on options for supporting multiple languages for your customers.

Order - Main - Account Information

The **Account Information** section contains account-oriented information.

> **CAUTION:** This page contains a subset of account-oriented information. If you need to define other account-oriented information, transfer to Account - Main after the order is completed.

Whether or not an account is required or even allowed on an order is controlled by an option on the **Campaign**. If the **Campaign** disallows the use of an account, this section will be suppressed. If the **Campaign** allows the use of an account, an indication of whether an account is optional or required is displayed adjacent to **Account Information**.

If an account is allowed on the order, use the dropdown at the top of this section to tell the system about this order's account:

- Select `No Account Information` if you do not want to associate an account with this order (e.g., for a person-oriented marketing survey).
- Select `Create New Account` if this is a new account. You should enter the new account's information in the remaining fields in this section.
- Select `Use Existing Account` if an existing account should be used on the order. If an existing person was selected above, one of the person's accounts (if any) will be automatically displayed in the **Account Information**. In addition, if the order references an existing person, the account search will be limited to accounts associated with this person. If an **Existing Account** is selected, the following takes place:
-
  - Basic information about the account appears beneath the search area (and the account context menu can now be used to drill down to account-oriented pages).
  - If any of the Questions & Misc Fields on the next tab are used to capture account-oriented characteristics, the account's existing characteristic values are defaulted.
  - Information about the account is displayed in the fields that follow.

Customer Class plays a part in:

- If and when a customer is subject to late payment charges.
- The account's default collection class and when the account debt monitor reviews an account. Refer to The Big Picture of Credit & Collections for more information about how and when an account's debt is reviewed.
- And several other functions. Refer to and Setting Up Customer Classes for more information.

> **Note:** For new accounts, **Customer Class** defaults based on the **Campaign**.

The optional Account **Management Group** controls the roles assigned to To Do entries associated with an account. Refer to Setting Up Account Management Groups for more information.

> **Note:** For new accounts, **Management Group** defaults based on the **Campaign**.

Copyright © 2024, Oracle and/or its affiliates.

The remaining fields are used to describe how the bill should be routed to the customer.

CAUTION: When an order is completed, several fields that do not appear on this page that reside on Account - Person are populated. For example, **Account Relationship Type, Bill Format**, **Number of Copies**, **Receives Notification** are all populated with default values. If you need to override the default values, transfer to Account - Person after the account is added.

Use **Bill Route Type** to define how the bill is sent to the customer. This field's value defaults from the Installation Record. This value is important as it controls many options on the remainder of this page. If the **Bill Route Type** indicates that bills are routed via Fax, the person's fax number is displayed adjacent (the system knows which of a person's phone numbers is a fax number by the **Phone Type**). If the **Bill Route Type** indicates that bills are routed via Email, the person's **Email** addressis displayed adjacent. If the **Bill Route Type** indicates that bills are routed via the Postal service, you must choose an appropriate **Bill Address Source** to define which address should be used. Refer to Setting Up Bill Route Types for more information about bill route types.

If the **Bill Route Type** indicates that bills are routed via the Postal service, you must choose the appropriate **Bill Address Source** to define which address should be used. This field will be protected for other **Bill Route Types**.

- Choose `Person` if bills should be sent to the *person's mailing address*. This address is displayed adjacent.
- Choose `Account Override` if bills should be sent to an override address specified below. Typically, you would only choose this option if the person has multiple accounts and each account's bills should be sent to a different address.

If you select a **Bill Address Source** of `Account Override`, you must enter the address to which bills will be sent in the following address constituents. The number and type of address constituents is based on the **Country** (refer to Defining Countries for more information on address constituents). These fields will be protected for other **Bill Address Sources**. Note, the **Country** defaults from Installation Options - System and several other constituents default based on the **Country** and **Postal Code** if a default postal exists for the postal code.

Order - Questions & Misc. Fields

This page has several purposes:

- The order's Campaign can define questions to be posed to customers when an order is taken. For example, you could ask the customer "marketing survey" questions (e.g., please rate our service, how do you heat your home?, etc.). This page allows you to capture the customer's response.
- The order's Campaign can also be setup to capture additional fields to be updated on the order's person / account when the order is completed. For example, you could indicate that the account's override due date and automatic payment options should be captured when an order is taken. This page allows you to capture these field values.
- The order's Campaign can also be setup to pose eligibility-oriented questions that qualify the customer for packages of goods and services. For example, you might have a campaign with packages that can only be selected if the customer commits to a year's worth of service. This page allows you capture the customer's response to these types of questions.
- After entering the above information (and the information on the Main page), you can use this page to select a package of goods and services for the customer.
- You can use this page to switch the campaign on the order (which would result in new questions and miscellaneous fields appearing).

Fastpath: Refer to The Big Picture of Campaigns, Packages and Orders for more information.

Open this page using **Sales & Marketing, Order** and then navigate to the **Questions & Misc Fields** tab.

Fastpath: Refer to Order User Interface Flow for information about how to use this page.

Description of Page

Copyright © 2024, Oracle and/or its affiliates.

Order contains a concatenation of the customer's name, campaign, package (if the order is completed), status and start date. **Order ID** is the system-assigned unique identifier of the order. These values only appear after the order is added to the database.

Status contains the state of the order. Refer to Lifecycle of an Order for information about the valid values and the state transition rules. Additional information may also be displayed based on the status:

- If the order's status is `Held`, the following information is also displayed: the hold reason, the name of the user who held the order, and the date and time when a reminder for the order is due to be created.
- If the order's status is `Canceled`, the following information is also displayed: the cancel reason, the name of the user who canceled the order, and the date and time when the order was canceled.
- If the order's status is `Complete`, the following information is also displayed: the name of the user who completed the order and the date and time when the order was completed.

Press the **Hold** button to hold a `Pending` order. Refer to Order - Hold Dialog for a description of the pop-up window used to hold an order.

Press the **Cancel** button to cancel a `Pending` or `Held` order. Refer to Order - Cancel Dialog for a description of the pop-up window used to cancel an order.

If the **Campaign** has been setup to allow the completion of orders without selecting a package, a button appears to the right of **Cancel**. The label that appears in the button is defined on the **Campaign**. When pressed, the order is completed. When the order is completed,

- The "V" is setup / updated.
- All questions and miscellaneous fields (defined on the next tab) are posted to the appropriate object.
- Optional plug-ins are executed.
- You will be transferred to the order completion transaction that's defined on the **Campaign** (if any).

If the order references an existing person or account and the related person / account has changed after their information was snapshot on the order, the **Out of Sync** section appears. This section highlights which objects have changed since the order was last saved. In addition, the **Sync** button appears. When pressed, the order is updated with the current field values for these objects.

The number and type of **Questions & Misc Fields** is controlled by the Campaign.

Note: Some questions can be suppressed. The campaign contains an indication if each question is account-oriented, person-oriented or order-oriented. If the question is account-oriented and no account is defined on the **Main** tab, the question will be suppressed.

Each row in this grid has a **Prompt** and a **Response**. The value of each **Prompt** is defined on the **Campaign**. The **Response** holds the answer supplied by the customer. Please note the following about the **Response** field:

- On a **Campaign**, you can indicate which questions / misc. fields are required. The Response for a required field is prefixed with an asterisk - `*`.
- The value of a Response can default from a variety of sources:
- - If an existing person was selected on the **Main** page AND the response is used to capture a person characteristic, the existing characteristic value will be defaulted.
  - If an existing account was selected on the **Main** page AND the response is used to capture an account characteristic, the existing characteristic value will be defaulted.
  - Otherwise, the characteristic value is defaulted from the **Campaign**.
- The type of field used to capture a Response is controlled by the Campaign. Depending on the type of Response you may see Search buttons.

After entering the **Responses** to the **Questions & Misc Fields**, press the **Show Eligibility** button.

The following may occur when this button is pressed:

- If a new customer is specified on the Main page, a warning will advise you that a new person will be added. The system needs to add a new person so that an audit of the customer interaction exists.

Copyright © 2024, Oracle and/or its affiliates.

- The tree expands to show the Order - Eligibility Tree containing package, ineligible package, error package, and other campaign nodes:
- • Each **Package** node contains a package that is linked to the campaign whose Eligibility Criteria allows its selection by the customer. You can expand this node to see the various contracts that will be created if you select the package. Selecting one of these packages causes the Order - Package Confirmation page to appear in which you can confirm your selection.
  - The **Ineligible Package(s)** node contains packages whose Eligibility Criteria prevents their selection by the customer.
  - The **Error Packages** node contains packages whose Eligibility Criteria have returned errors (if any).
  - The **other campaign(s)** node contains alternative campaigns that may be substituted for the current campaign on the package. Selecting one of these campaigns will cause the campaign to be changed on this order. Changing a campaign on an order will result in new questions and miscellaneous fields appearing. Refer to The Big Picture Of Campaign Eligibility for more information.

Note: Some nodes can be suppressed. To increase page performance with large campaigns, you can define a feature configuration to suppress the display of the Ineligible Packages, Error Packages or Other Campaign nodes. If a node is suppressed, it does not appear on the eligibility tree and the system does process the information necessary to display the node.

Order - Log

This page contains an audit trail of significant events that transpired in the history of this order.

Open this page using **Sales & Marketing, Order** and then navigate to the **Log** tab.

Description of Page

Order contains a concatenation of the customer's name, campaign, package (if the order is completed), status and start date. **Order ID** is the system-assigned unique identifier of the order. These values only appear after the order is added to the database.

Status contains the state of the order. Refer to Lifecycle of an Order for information about the valid values and the state transition rules. Additional information may also be displayed based on the status:

- If the order's status is `Held`, the following information is also displayed: the hold reason, the name of the user who held the order, and the date and time when a reminder for the order is due to be created.
- If the order's status is `Canceled`, the following information is also displayed: the cancel reason, the name of the user who canceled the order, and the date and time when the order was canceled.
- If the order's status is `Complete`, the following information is also displayed: the name of the user who completed the order and the date and time when the order was completed.

Press the **Hold** button to hold a `Pending` order. Refer to Order - Hold Dialog for a description of the pop-up window used to hold an order.

Press the **Cancel** button to cancel a `Pending` or `Held` order. Refer to Order - Cancel Dialog for a description of the pop-up window used to cancel an order.

Campaign contains basic information about the order's campaign.

Package only appears if the order has been completed. It contains basic information about the package.

Start Date contains the date on which the contract's created by the order will be started.

The **Log** contains an audit trail of significant events that transpired in the history of this order. There are two types of entries in the log:

- The system automatically adds log entries when key events occur. These entries may not be removed or changed.
- Users may add log entries by pressing the + button and then specifying the Details.

Copyright © 2024, Oracle and/or its affiliates.

Order - Package Confirmation

The Package Confirmation page allows you to check the package you selected before the V is created / updated for the customer. This page opens when you select a package from the eligibility tree on Order - Questions & Misc Fields.

> **CAUTION:** If you press the **Complete** button, a variety of objects may be created (contracts, an account, etc.). If you need to change any of these objects after order completion, you must transfer to the respective page on which the object is maintained. You can also use Start / Stop to make changes to the contracts created by this transaction.

Description of Page

Order contains a concatenation of basic information about the order. **Order ID** is the system-assigned unique identifier of the order. These values only appear after the order is added to the database.

Campaign contains basic information about the order's campaign.

Package contains basic information about the package.

Person is the name of the customer.

Account is the name of the customer and their customer class.

Start Date contains the date on which the contract's created by the order will be started.

The **Questions & Misc Fields** grid contains the questions and miscellaneous fields that appeared on the order. Note well, it is possible to design a campaign and a package where some fields are optional on the campaign but required for a specific package. The **Response** of all required fields for the selected package are prefixed with an asterisk. Refer to Order - Questions & Misc Fields for information about the rows in this grid.

> **Note: Changing your answers could cause the package to become ineligible.** As described earlier, the fields you enter on an order play are what the system uses to determine if a package can be selected for a customer. If you change the answers to the **Questions & Misc Fields**, the package may no longer be eligible and an error will be presented to you. You will have the choice of reverting to the original answers or returning to the order page where you can select an eligible package.

> **Note: Additional questions / misc fields are possible.** It is possible to design a campaign where certain field values are only visible on specific packages. In this scenario, the **Questions & Misc Fields** grid may contain additional rows for a given package. Refer to Designing Campaigns and Packages for more information.

The **Instructions** grid contains package-specific suggestions. These instructions are defined on the package.

The **Messages** grid contains system-generated messages that summarize what will happen if you press the **Complete** button.

Press the **Complete Order** button if you are satisfied with the information on the order. Otherwise, press **Do Not Use this Package** to change or cancel the order.

Order - Hold Dialog

The **Order - Hold** dialog opens if you press the **Hold** button on the order transaction. You use it to define why the order is being held and when you want to be reminded to check up on the order.

Description of Page

Use **Hold Reason** to define why the order is being held.

The following fields are used to control the To Do entry that will be created to remind you about this order.

- Use **Reminder Date / Time** to define the *latest* date on which the To Do entry should be created. The reason we indicated this should be the *latest* date is because the background process (known by the batch control ID of `TD-`

Copyright © 2024, Oracle and/or its affiliates.

ECBK) that's responsible for creating these To Do entries has a parameter called "lead time". This parameter is used to define the number of days before the **Reminder Date** that the To Do entry should be created.

- If the To Do entry should be addressed to a group of users, choose a **Reminder Type** of `Send to Role` and enter the user group's **To Do Role**. Note, the To Do Role will default from the order's campaign.
- If the To Do entry should be addressed to a specific user, choose a **Reminder Type** of `Send to User` and enter the user's User ID in **To Do User**.
- **Log Notes** are used to describe why the order is being held. This information appears in the log entry that is created by the system when you hold an order.

If there are any To Do entries (with a status of `Open` or `Being Worked On`) that drill down to this order, a message is displayed indicating that existing To Do entries for this order will be completed. If the order is already in `Hold` status, existing To Do entries are completed when you click the OK button and a new To Do entry is created when the background process responsible for creating the To Do entries runs.

Order - Cancel Dialog

The Order - Cancel dialog opens if you press the **Cancel** button on the order transaction. You use it to define why the order is being canceled.

Description of Page

Use **Cancel Reason** to define why the order is being canceled.

Use **Log Notes** to describe why the order is being canceled. This information appears in the log entry that is created by the system when you cancel an order.

Maintaining Campaigns

When you add an order, you must reference its campaign. An order's campaign controls:

- The type of information defaulted onto an order. For example, you can setup a campaign to default a given account management group on all orders linked to this campaign. This account management group will then default onto accounts created when the campaign's orders are completed.
- The eligibility-oriented questions that are posed to the customer when an order is taken. For example, the questions indicated above - Would you like to pay automatically? / What is your date of birth? - are all defined on the order's campaign.
- The superset of packages that can be offered to a customer whose order references this campaign. An order's campaign defines the types of packages that may be selected.
- In addition to the above, campaigns also control high-level *eligibility rules* and business process flows.

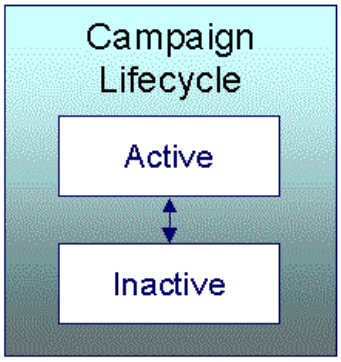
The topics in this section describe the campaign transaction.

Fastpath: Refer to The Big Picture of Campaigns, Packages and Orders for general information about campaigns. Refer to The Big Picture Of Campaign Eligibility Rules for a discussion of how to control which campaigns can be used for different types of customers. Refer to Designing Campaigns and Packages for guidelines describing how to design your campaigns.

Lifecycle of a Campaign

The following diagram illustrates the lifecycle of a campaign.

Copyright © 2024, Oracle and/or its affiliates.



`Active` An `Active` campaign may have orders created in respect of it.

`Inactive` By `Inactivating` the campaign, you are preventing it from being used on future orders.

Campaign - Main

This page is used to define basic information about a campaign.

Fastpath: Refer to The Big Picture of Campaigns, Packages and Orders and Designing Campaigns and Packages for more information.

Open this page using **Sales & Marketing, Campaign**.

Description of Page

Campaign is the user-defined code that uniquely identifies the campaign. The campaign's description appears adjacent.

Description briefly describes the campaign.

Campaign Status describes the state of the campaign. Refer to Lifecycle of a Campaign for more information.

The following fields contain values to be defaulted onto this campaign's orders. The user who takes the order may override these default values. Note well, these default values are only applicable for new persons - they will not override field values defined on existing persons.

- The system requires all persons to be defined as either a `Person` or a `Business`. Use **Default Person / Business** to define the default value for persons created by this campaign's orders.
- Every phone number defined for a person must reference a phone type. Use **Default Phone Type 1** and **Default Phone Type 2** to define the first two phone types defaults for persons created by this campaign's orders.

Account Usage controls if and how this campaign's orders use accounts. You have the following options:

- Select `Not Allowed` if this campaign's orders do not reference an account.
- Select `Optional` if this campaign's orders do not have to reference an account. This option also means that a new OR existing account may be specified on the order. Note, if an order references a new account, the account will be created when the order is completed.
- Select `Optional - Existing Account Only` if this campaign's orders do not have to reference an account. This option also means that only an existing account may be specified on an order.
- Select `Required` if this campaign's orders must reference an account. This option also means that a new OR existing account may be specified on the order. Note, if an order references a new account, the account will be created when the order is completed.
- Select `Required - Existing Account Only` if this campaign's orders must reference an account. This option also means that only an existing account may be specified on an order.

The following fields contain values to be defaulted onto this campaign's orders. The user who takes the order may override these default values. Note well, these default values are only applicable for new accounts - they will not override field values defined on existing accounts.

Copyright © 2024, Oracle and/or its affiliates.

- The system requires all accounts to belong to a customer class. Use **Default Customer Class** to define the default value for accounts created by this campaign's orders.
- An account may reference an account management group. Use **Default Acct Management Group** to define the default value for accounts created by this campaign's orders.

If a user holds one of the campaign's orders, they can have a To Do entry generated on a future date to remind them about the order. Use **Default Hold To Do Role** to define the group of users to whom the To Do entry will be sent. Note well, the user who holds an order can indicate that the related To Do entry should be addressed to a specific user ID rather than a **Role**.

Use **Order Open Option** to define the tab page that is displayed when a user adds a new order for this campaign. This field is only relevant when the order transaction is invoked in "add mode" and a campaign is passed to this transaction.

A user can be automatically transferred to a different transaction when they complete an order. Refer to Supported Business Processes for examples of transactions that we anticipate being used. Use **Navigation Option** to define this transaction. Note,

- Leave **Navigation Option** blank if the user should remain on the order transaction after completion.
- You can setup **Navigation Options** to transfer to the user to a tab other than the main tab.

If a user is allowed to complete this campaign's order's without selecting a package,

- Turn on **Allow Order Compl w/o Package** and
- Use **Order Compl w/o Package Label** to define the message to appear in the button used to complete such orders.

Fastpath: Refer to Campaigns Without Packages for examples of campaigns where you would enable the above fields.

Use **Long Description** to provide an overview of this campaign.

The grid at the bottom of the page summarizes this campaign's questions and miscellaneous fields. To change this information, press the adjacent go to button (which will transfer you to the Questions and Misc Fields tab). To add additional information, transfer to the Questions and Misc Fields tab and press the + button.

Campaign - Questions and Misc. Fields

This page is used to define the following type of information:

- An order's campaign can define questions to be posed to customers when an order is taken. For example, you could ask the customer "marketing survey" questions (e.g., please rate our service, how do you heat your home?, etc.).
- An order's campaign can also be setup to capture additional fields to be updated on the order's person / account when the order is completed. For example, you could indicate that the account's override due date and automatic payment options should be captured when an order is taken.
- An order's campaign can also be setup to pose eligibility-oriented questions that qualify the customer for packages of goods and services. For example, you might have a campaign with packages that can only be selected if the customer commits to a year's worth of service.

Fastpath: Refer to Designing Campaigns and Packages for information describing how to design this type of information for your campaigns.

Open this page using **Sales & Marketing, Campaign** and navigate to the **Question & Misc Fields** tab.

Description of Page

CAUTION: This information is not intuitive; we strongly recommend that you follow the guidelines under Designing Campaigns and Packages before attempting to enter this information.

Campaign is the user-defined code that uniquely identifies the campaign. The campaign's description appears adjacent.

The **Questions & Misc Fields** scroll contains one row for each question / miscellaneous field captured on this campaign's orders. The following fields are used:

Copyright © 2024, Oracle and/or its affiliates.

- **Prompt on Order** contains the prompt that is shown to the user when the order is taken. Refer to Determine The Properties Of Every Miscellaneous Field for examples.
- **Brief Description** is simply a brief description of the question / miscellaneous field.
- **Sort Sequence** controls the order in which the question appears on the order.
- **Applicability** controls whether the field is `Optional`, `Required`, or `Only Applicable on Package`.
-
  - A value of `Required` means the order taker must enter a valid value before packages are displayed for selection.
  - A value of `Optional` means the order taken can leave the field blank prior to package selection. You can make an `Optional` field `Required` on individual packages.
  - A value of `Only Applicable on Package` is used for fields that are only captured for specific packages. Fields of this type are suppressed prior to package selection. When you setup a package, you indicate if the field is applicable (if so, the user is only prompted for the field if the package is selected).
- **Type of Response** controls where the field is stored when the order is completed and how it is validated. You have two options: `Characteristic` or `Column`.
-
  - If a field is a `Characteristic`, you must define which entity the characteristic value is to be stored on when the order is completed (`Person`, `Account`, or `Order`). You must also define the **Characteristic Type** as this controls the characteristic value entered by the user.
  - If a field's value resides in a `Column` (as opposed to a characteristic value), you must define the column reference. A field's **Column Reference** controls three things:
- How the field's value is retrieved. A field's value is retrieved when an existing person / account is referenced on an order.
- How the field's value is validated. A field's value is validated before package eligibility is determined and before the order is completed.
- How the field's value is posted (i.e., updated on the database). A field's value is posted when an order is completed.

Fastpath: Refer to *Extract Column References From Miscellaneous Fields* for more information about column references.

- If a field's value resides in a `Column` (as opposed to a characteristic value), you must also define whether the question / field is dependent on the order referencing an account. You do this by populating **Question Dependency** with one of the following values:
- Select `Must Have Account` if this question / field should only appear if the order references an account (new OR existing).
- Select if this question / field should only appear if the order references (new OR existing).
- Select `No Dependency` if this question / field should always appear.

Fastpath: Refer to Determine Account Dependencies For Column References for more information.

- **Default Column Value**/ **Default Char Value** controls whether the order taker is supplied with a default value when an order is created. Note well: for `Optional` and `Only Applicable on Package` fields, you can specify a different default value on each package.

Campaign - Algorithm

This page defines plug-in algorithms that are executed for orders associated with this campaign. Open this page using **Sales & Marketing, Campaign** and navigate to the **Algorithm** tab.

Description of Page

Campaign is the user-defined code that uniquely identifies the campaign. The campaign's description appears adjacent.

The grid contains **Algorithms** that control campaign-oriented functions. You must define the following for each plug-in algorithm:

- Specify the **System Event** with which the algorithm is associated (see the table that follows for a description of all possible events).

Copyright © 2024, Oracle and/or its affiliates.

- Specify the **Sequence** and **Algorithm** for each **System Event**. You can set the **Sequence Number** to 10 unless you have a **System Event** that has multiple **Algorithms**. In this case, you need to tell the system the **Sequence** in which they should execute.

The following table lists and describes the system events to which you can attach algorithms in a campaign:

System Event	Optional / Required	Description
`Campaign Eligibility`	Required	A campaign's eligibility algorithms control: - Whether the campaign appears on an order's eligibility tree. Campaigns that appear in this tree are alternate campaigns that may be used on an order. Refer to Real-time Marketing of Services to a Prospect for an example of how an order's eligibility tree can contain alternate campaigns for an order. Refer to The Big Picture Of Campaign Eligibility for information about these algorithms. Click here to see the algorithm types available for this system event.
`Order Completion`	Optional	When an order is completed for this campaign, algorithms plugged into this spot are called to do additional work (e.g., create a customer contact). You need only specify this type of algorithm if you require additional work to be performed when an order is completed for this campaign. Click here to see the algorithm types available for this system event.

Campaign - Package

A package controls the various types of contracts that will be created if the customer selects the package. This page summarizes all packages that may be selected on orders associated with this campaign. Open this page using **Sales & Marketing, Campaign** and navigate to the **Package** tab.

Description of Page

Campaign is the user-defined code that uniquely identifies the campaign. The campaign's description appears adjacent.

The grid contains an entry for every package associated with this campaign. To view or change a package, press the adjacent go to button (which will transfer you to Package - Main). To add a new package, use the **Campaign** context menu to transfer to Package - Main in add mode.

Maintaining Packages

After a user enters basic information about a customer on an order, they are shown the various packages that may be offered to the customer. A package controls the various types of contracts that will be created if the customer selects the package.

If the customer elects to take a package, the system sets up / updates the "V" (along with all of the ancillary things that happen when service is initiated).

The topics in this section describe the package transaction.

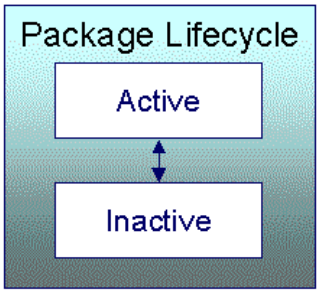
Copyright © 2024, Oracle and/or its affiliates.

Fastpath: Refer to The Big Picture of Campaigns, Packages and Orders for general information about packages. Refer to The Big Picture Of Package Eligibility Rules for a discussion of how to control which packages can be used for different types of customers. Refer to Designing Campaigns and Packages for guidelines describing how to design your packages.

Note: An order can be completed without selecting a package. It is possible to use the order transaction to simply create / update persons and accounts. Refer to Campaigns Without Packages for more information.

Lifecycle of a Package

The following diagram illustrates the lifecycle of a package.



`Active` An `Active` package may be selected on an order.

`Inactive` By `Inactivating` a package, you are preventing it from being selected on future orders.

Package - Main

This page is used to define basic information about a package.

Fastpath: Refer to The Big Picture of Campaigns, Packages and Orders and Designing Campaigns and Packages for more information.

Open this page using **Sales & Marketing, Package**.

Description of Page

Package contains a concatenation of basic information about the package. **Package ID** is the system-assigned unique identifier of the package. These values only appear after the order is added to the database.

Use **Campaign** to define the campaign to which this package belongs.

Use **Description** and **Long Description** to describe the package.

Status defines the state of the package. Refer to Lifecycle of a Package for information about the valid values and the state transition rules.

The tree at the bottom of the page summarizes the information that appears on the remaining tabs. You can click on a hyperlink to transfer you to the appropriate tab with the relevant information displayed.

Package - Contracts To Create

This information on this page controls the type of contract(s) that are created if this package is selected on an order.

Fastpath: Refer to The Big Picture of Campaigns, Packages and Orders and Designing Campaigns and Packages for more information.

Copyright © 2024, Oracle and/or its affiliates.

Open this page using **Sales & Marketing, Package** and navigate to the **Contracts To Create** tab.

Description of Page

Package contains a concatenation of basic information about the package. **Package ID** is the system-assigned unique identifier of the package. These values only appear after the order is added to the database.

CAUTION: This following information is not intuitive; we strongly recommend that you follow the guidelines under Designing Campaigns and Packages before attempting to enter this information.

The **Contracts To Create** scroll contains one row for every contract that will be created if this package is selected on an order. The following fields are used:

- **Division** and **Contract Type** define the *type of contract* to be created.
- **Start Method Algorithm** controls how the system creates the new contract.
- **Create Proposal Contract** controls if the new contract is a real or a proposal contract. This switch only appears if the quotation `Contract Management` module is not turned off. Refer to *Proposal Contracts Must Exist* for more information.

Package - Questions & Misc Fields

It is possible to design a campaign so that the user is prompted for additional information when a package is selected on an order. This page is used to define how the package overrides the questions and miscellaneous fields defined on the campaign.

Fastpath: Refer to The Big Picture of Campaigns, Packages and Orders and Designing Campaigns and Packages for more information.

Open this page using **Sales & Marketing, Package** and navigate to the **Questions & Misc Fields** tab.

Description of Page

Package contains a concatenation of basic information about the package. **Package ID** is the system-assigned unique identifier of the package. These values only appear after the order is added to the database.

Campaign contains a concatenation of basic information about the campaign.

CAUTION: The following information is not intuitive; we strongly recommend that you follow the guidelines under Designing Campaigns and Packages before attempting to enter this information.

The grid contains one row for every Questions & Misc Field defined on the **Campaign**. You need only modify this information if this package behaves differently than the campaign (e.g., a field is required on the package, but optional on the campaign). To override a field on this package, turn on the **Override** switch and change the appropriate fields. There are several restrictions in respect of what can be overridden as described below:

- If a question / miscellaneous field is `Required` on the campaign, no fields may be overridden on the package.
- If a question / miscellaneous field is `Optional` on the campaign, you may override all field values. However, if the package has been used on a `Completed` order, you may not change its **Applicability** to be `Required`.
- If a question / miscellaneous field is designated as `only applicable on package` on the campaign, you should designate its applicability as `Required`, `Optional` or `Not Used On This Package`. If the field is either `Required` or `Optional` on this package, you may also override the other field values. Please note that if the package has been used on a `Completed` order, you may not change its **Applicability** to be `Required`.

Fastpath: Please refer to the `Description of Page` that appears under Campaign - Questions & Misc Fields for a description of the fields on the page.

Copyright © 2024, Oracle and/or its affiliates.

Package - Eligibility

This page is used to define the conditions under which a package can be selected on an order.

Fastpath: Refer to The Big Picture Of Package Eligibility for more information.

Open this page using **Sales & Marketing, Package** and navigate to the **Eligibility** tab.

Description of Page

Package contains a concatenation of basic information about the package. **Package ID** is the system-assigned unique identifier of the package. These values only appear after the order is added to the database.

CAUTION: The following information is not intuitive; we strongly recommend that you follow the guidelines under The Big Picture Of Package Eligibility before attempting to enter this information.

The **Eligibility Criteria Group** scroll contains one entry for each group of eligibility criteria. The following fields may be defined for each group:

- Use **Sort Sequence** to control the relative order in which the group is executed when the system determines if the package can be selected on an order (smaller numbers are executed before larger numbers).
- Use **Description** and **Long Description** to describe the criteria group.
- Use **If Group is True** to define what should happen if the eligibility criteria (defined in the following grid) return a value of `True`.
- - Choose `Eligible` if this package can be selected on the order.
  - Choose `Ineligible` if this package cannot be selected on the order.
  - Choose `Check Next Group` if the next criteria group should be checked.
- Use **If Group is False** to define what should happen if the eligibility criteria (defined in the following grid) return a value of `False`.
- - Choose `Eligible` if this package can be selected on the order.
  - Choose `Ineligible` if this package cannot be selected on the order.
  - Choose `Check Next Group` if the next criteria group should be checked.

The grid that follows contains the package's eligibility criteria. Think of each row as an "if statement" that can result in the related eligibility group being True or False. For example, you might have a row that indicates the customer is eligible for the package if their customer class is financial service. The following bullets provide a brief description of each field on an eligibility criterion. Please refer to Defining Logical Criteria for several examples of how this information can be used.

- Use **Sort Sequence** to control the order in which the criteria are checked.
- Use **Criteria Field** to define the field to compare:
- - Choose `Field` if you want to compare a response to a question / miscellaneous field to a given value. Push the adjacent search button to select the field.
  - Choose `Algorithm` if you want to compare anything other than a response to a question / miscellaneous field. Push the adjacent search button to select the algorithm that is responsible for retrieving the comparison value.
- Use **Criteria Comparison** to define the method of comparison:
- - Choose `Algorithm` if you want an algorithm to perform the comparison and return a value of True, False or Insufficient Data.
  - Choose any other option if you want to compare the **Criteria Field** using a logical operator. The following options are available:
- Use >, <, =, >=, <=, <> (not equal) to compare the **Criteria Field** using standard logical operators. Enter the comparison value in the adjacent field.
- Use `IN` to compare the **Criteria Field** to a list of values. Each value is separated by a comma. For example, if a field value must equal `1`, `3` or `9`, you would enter a comparison value of `1,3,9`.
- Use `BETWEEN` to compare the **Criteria Field** to a range of values. For example, if a field value must be between `1` and `9`, you would enter a comparison value of `1,9`. Note, the comparison is inclusive of the low and high values.

Copyright © 2024, Oracle and/or its affiliates.

- The next three fields control whether the related logical criteria cause the eligibility group to be considered True or False:
- - Use **If TRUE** to control what happens if the related logical criterion returns a value of True. You have the options of `Group is true`, `Group is false`, or `Check next condition`. If you indicate `Group is true` or `Group is false`, then the package will be judged `Ineligible` or `Eligible` based on the values defined above in **If Group is False** and **If Group is True**.
  - Use **If FALSE** to control what happens if the related logical criterion returns a value of False. You have the options of `Group is true`, `Group is false`, or `Check next condition`. If you indicate `Group is true` or `Group is false`, then the package will be judged `Ineligible` or `Eligible` based on the values defined above in **If Group is False** and **If Group is True**.
  - Use **If INSUFFICIENT DATA** to control what happens if the related logical criterion returns a value of "Insufficient Data". You have the options of `Group is true`, `Group is false`, or `Check next condition`. If you indicate `Group is true` or `Group is false`, then the package will be judged `Ineligible` or `Eligible` based on the values defined above in **If Group is False** and **If Group is True**.

Package - Instructions

This page is used to define special instructions that should be presented to the user when they select this package. This information appears on the Order - Package Confirmation page.

Fastpath: Refer to Determine If Additional Instructions Are Needed for more information.

Open this page using **Sales & Marketing, Package** and navigate to the **Instructions** tab.

Description of Page

Package contains a concatenation of basic information about the package. **Package ID** is the system-assigned unique identifier of the package. These values only appear after the order is added to the database.

The grid contains a row for each instruction to be presented to the user when they select a package. The following fields are used:

- Use **Sort Sequence** to control the relative order of the instruction.
- Use **Display Icon** to control the icon that prefixes the instruction. Refer to Display Icons for more information.
- Use **Description** to define the text of the instruction.

Chapter 9

Rates

Topics:

- All Rates Share A Common Structure
- How To Create A New Rate
- Control Tables That Must Be Set Up Before Creating A Rate Schedule
- Defining Service Quantity (SQ) Identifiers
- Setting Up A Rate Schedule
- Rate Schedule Merge
- Setting Up Bill Factors
- Defining Rate Versions
- Designing Rate Components
- Defining Rate Components
- Rate Version Merge
- Rate Check
- Rate Schedule Summary
- Effective Dates & Price Proration
- How To Add A New Service Quantity Rule Algorithm
- How To Add A New Register Rule Algorithm

When you set up a customer in the system, you create a contract for every service that you sell to the customer. The most important information specified on a contract is the rate. The rate controls:

- How the service's charges are calculated.
- How the charges appear aesthetically on the customer's bills.
- How the general ledger is affected by the charges.

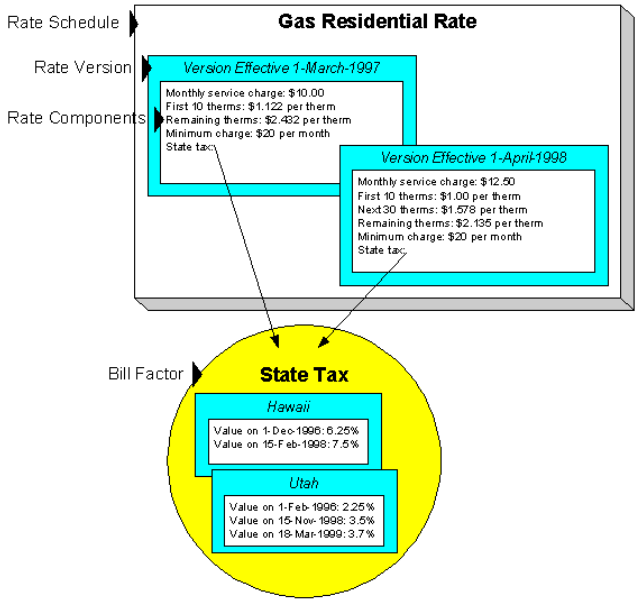
In this section, we explain how to set up your organization's rates.

Fastpath: Refer to How Rates Affect The Information On A Bill for how billing uses the information in a rate to calculate a bill segment.

Note: Not all contracts reference a rate. It's important to understand that you may have contracts that don't use a rate to calculate their charges. For example, you may have billable charge contracts that exist to bill a customer for charges that have been calculated in a foreign system. Whether or not a contract uses a rate and the list of rates that can be specified on a contract are controlled by the contract's contract type.

All Rates Share A Common Structure

All rates share a common structure as illustrated in the following sample rate:



This rate is constructed using the following components:

- Every rate has a single **rate schedule** that contains information about the rate that doesn't change over time. For example, the rate's description.
- A rate's effective-dated calculation algorithms are stored in a **rate version**. Every rate schedule has at least one rate version. Multiple rate versions exist when something about the algorithms changes and it's important to keep the prior version in order to recalculate historical bills. The rate shown above has 2 rate versions; one effective on 1-March-1997, the other effective on 1-April-1998.
- A rate version's calculation algorithms are stored in **rate components**. Every rate version will have at least one rate component. The number of rate components linked to a rate version is dependent on the complexity of the calculation rules. We have seen very simple calculation rules that need only a single rate component; we have seen others that require over 150 rate components. You will find the construction of rate components to be the most challenging task of the setup process.
- **Bill factors** are used to specify the amount to charge when the amount is the same for many rates. Notice that both of the above rate versions reference the prevailing "state tax" bill factor rather than specifying the specific tax percent. This allows the tax rate to be maintained in one place.

> **Note: Bill factors have many uses.** Bill factors have many uses in addition to specifying tax rates. Refer to An Overview Of Bill Factors for all the details.

The remainder of this chapter discusses the intricacies of setting up the rates and bill factors.

How To Create A New Rate

The following points describe steps you'll follow to set up a rate.

- Create a *rate schedule* for the rate. Refer to Setting Up A Rate Schedule for more information.
- Set up a *rate version* to establish the earliest effective date of the rate. Refer to Defining Rate Versions for more information. We recommend that you set the status of the rate version to `In Progress` while you're working on the rate.

Copyright © 2024, Oracle and/or its affiliates.

- Set up rate components to define the rate version's calculation rules. Refer to Defining Rate Components and Rate Version Merge for more information.
- After all rate components have been setup, return to Rate Version - Main and change the rate version's status to `Validated`. At this point, you can use Rate Check to test the rate version.
- After you have checked the rate, return to the Rate Version - Main and change the status to `Finished`. This allows the rate to be used by billing.
- Define the various contract types that can use the rate. You do this by referencing the rate on *Contract Type - Rate*.

Control Tables That Must Be Set Up Before Creating A Rate Schedule

This section describes tables that must be set up before you can add a new rate schedule.

Defining Frequency Codes

When you create a rate schedule, you must enter a code defining the frequency in which the prices are expressed, e.g., monthly, quarterly, biannual, etc. To set up frequency codes, open **Admin Menu, Frequency**.

Description of Page

Enter a unique **Frequency** ID and **Description** for each frequency.

Before entering the remaining fields, you must analyze the "proration provisions" in your rates. For example, consider a rate used to calculate charges for a "month's worth of service". Assume this rate's charges will not be prorated when the customer's service period is anywhere between 27 and 33 days. This rate requires a frequency that indicates:

- `12` **Periods / Year** (because it's a monthly rate).
- **Minimum Days Offset** equal to `3`.
- **Maximum Days Offset** equal to `3`.

When the system creates a bill using a rate with this frequency, it compares the number of days of consumption on the bill to the minimum and maximum number of days inferred by the rate's frequency. In the example above, the minimum number of days would be 27 (365 days / 12 period per year - 3 offset for minimum days). The maximum number of days would be 33 (365 days / 12 periods per year + 3 offset for maximum days). Therefore, if a bill's consumption period is less than 27 days or more than 33 days, the bill's charges are prorated.

Note: Overriding the system's standard proration logic. If the standard proration logic does not satisfy your requirements, you may plug in an override proration algorithm to calculate the proration factors as required by your implementation. For example, we have encountered companies who treat a year has have 360 days rather than 365 days. Such a company would need to develop an override proration algorithm and plug it on the Installation record.

Where Used

Every Rate Schedule must have a frequency. Refer to Rate Schedule - Main for more information.

UOM versus Variance Parameter versus SQI

Almost all rates contain prices in respect of some number of something. This something could be anything from a therm of gas to a kilowatt-hour of electricity. We use three distinct codes to identify the "thing" being priced in a rate. These codes are **UOM** (unit of measure), **Variance Parameter**, and **SQI** (service quantity identifier). These codes must be set up before you can create a rate schedule.

You don't have to use all three codes when you specify a price in a rate. Rather, you use the codes that are needed to clearly identify the thing being priced. We'll use a few examples to clarify this concept.

Copyright © 2024, Oracle and/or its affiliates.

Let's consider a rate that simply has a price per therm. For this rate, you'd only use a UOM code to identify the thing being priced (and this code should be something simple like `THRM`).

Next, let's consider a rate that has a different price for kilowatts consumed during peak hours as compared to those used during off-peak hours. You would use a combination of UOM and Variance Parameter to define the thing being priced. In this example, we'd recommend using a single UOM for kilowatt-hours (labeled `KWH`) and two Variance Parameter - one for the peak period (labeled `PEAK`) and the other for the off-peak period (labeled `OFFPEAK`).

Note: Rule. If the thing being priced is measured by a meter, you must use the same UOM / Variance Parameter in the rate as are specified on the meter's register(s). Refer to *Setting Up Unit Of Measure Codes* for more information.

And finally, let's consider a rate that has a charge for "annual highest KW (demand) during peak hours". This is a very complicated thing to price. You have several options in respect of the combination of UOM, Variance Parameter, SQI to use to express this unit.

The following table illustrates various combinations of the UOM, Variance Parameter, and SQI to rate the charge:

Option	UOM	Variance Parameter	SQI
1	KW	PEAK	ANNHIGH - Annual Highest
2	KW		PEAKANHI - Peak Annual Highest
3			ANHIPKKW - Annual Highest Peak KW
4	KWAP - KW Annual Highest Peak		
5	KW	AHPEAK - Annual Highest Peak	

Note: Rule Of Thumb. We strongly recommend using an SQI to identify things not measured by your meters. This means that you shouldn't use the last two options in the above table. Of the first 3 options, the choice is yours. It's really a question of how you prefer to record your billed consumption.

Fastpath: For more information about UOM, Variance Parameter, and SQI, refer to *Setting Up Unit Of Measure Codes*, *Setting Up Variance Parameter*, and Defining Service Quantity (SQ) Identifiers.

Setting Up Register Rules

To set up register rules, choose **Admin Menu, Register Rule**.

Description of Page

Enter a unique **Register Rule** ID and a **Description** for the register rule.

Enter **Comments** that describe, in detail, what the rule does.

Select the **Algorithm** that controls how the system manipulates the consumption supplied to the rate. Define the **Parameters** to be used by the register algorithm.

The following table indicates the algorithms that are supplied with the system, its Algorithm Type and how each register algorithm uses its parameters.

Copyright © 2024, Oracle and/or its affiliates.

Base Package Register Rules

Note: If the register rule algorithms that are supplied with the base package are not sufficient for your rating requirements, you will have to write a new register rule algorithm. Refer to How To Add A New Register Rule for how to do this.

The following table lists and describes the register rule algorithms shipped with the product:

Register Algorithm	Alg Type	What It Does	Parameters Used
`Bill Factor Based UOM Conversion`	BU	Converts a measured quantity into a final quantity using the value specified on a bill factor. Final Quantity (expressed in the UOM in param 2) = Quantity for UOM (in param 1) * Value for Bill Factor (in param 3) This algorithm would be useful when you need to convert cubic feet (CCF) into therms based on the takepoint from which the location gets its gas.	Param 1 = Measured UOM Param 2 = Final UOM Param 3 = Bill Factor containing conversion factor
`Real Time Pricing`	RT	Converts a measured quantity (UOM) into a final quantity (Result SQI) by multiplying the measured quantity by the price (Bill Factor) effective on the read's start date/time. Final Quantity (expressed in the SQI in param 3) = Quantity for UOM (in param 1) * Value for Real Time Bill Factor (in param 2) Note, this option is useful if you have simple real time pricing algorithms and do not use the interval billing subsystem. Refer to *Pricing Interval Consumption Using Interval Prices* for more information.	Param 1 = UOM Param 2 = Bill Factor Param 3 = Resultant SQI
`Seasonal Register TOU Conversion`	ST	Converts Variance Parameter of "current season" and "prior season" to "winter"	Param 1 = Summer Begin Date (MMDD)

Copyright © 2024, Oracle and/or its affiliates.

Register Algorithm	Alg Type	What It Does	Parameters Used
		and "summer" based on the billing period. **Summer Begin Date** contains the MMDD when summer starts. **Winter Begin Date** contains the MMDD when winter starts. **Current Season Variance Parameter** contains the Variance Parameter of the register that measures the current season's consumption. **Prior Season Variance Parameter** contains the Variance Parameter of the register that measures the prior season's consumption. **Winter Variance Parameter** contains the Variance Parameter to be used for winter consumption (the register read's Variance Parameter is changed to this for winter consumption). **Summer Variance Parameter** contains the Variance Parameter to be used for summer consumption (the register read's Variance Parameter is changed to this for summer consumption).	Param 2 = Winter Begin Date (MMDD) Param 3 = Current Season TOU Param 4 = Prior Season TOU Param 5 = Winter TOU Param 6 = Summer TOU

For more information about defining algorithm types, refer to *Setting Up Algorithm Types*.

Where Used

A Rate Schedule may have zero or more register rules. Refer to Rate Schedule - Register Rules for more information.

Setting Up Service Quantity (SQ) Rules

To set up Service Quantity (SQ) rules, choose **Admin Menu, Service Quantity Rule**.

Description of Page

Enter a unique **SQ Rule** ID and a **Description** for the SQ rule.

Enter a **Long Description** that describes, in detail, what the rule does.

Indicate the **UOM, Variance Parameter** and **SQI** to be output by the rule.

Every service quantity supplied to a rate has two quantities - the amount of service as it was initially measured, and the amount of service that should be priced by the rate. Most of the time, these two amounts will be the same. They will only differ when another SQ Rule has executed and caused the billable quantity to change (an SQ Rule NEVER changes initially measured quantities). Because these two amounts may differ, an SQ Rule needs to know if it should use the

Copyright © 2024, Oracle and/or its affiliates.

quantity as initially measured OR if it should use the quantity that will be priced in the rate. You use the **SQ Quantity to Use** field to indicate such. Valid values are `Use Initial Quantity` and `Use Billable Quantity`.

Indicate whether the system should **Create Billing Error** if the SQ Rule is not able to execute successfully. For most SQ Rules you would turn this option on. However, if the rule is linked to a rate where the rule is only applicable to a subset of the customer base using the rate, you may not want a billing error to be issued if the rule cannot execute. For example, consider a rule used to derive total kWh by adding together peak kWh and off-peak kWh. You would only use such a rule if you measured peak and off-peak consumption, but you had prices in respect of total consumption. In the situation where this SQ Rule was linked to a rate used by customers with and without Variance Parameter meters, you would not want the system to issue a billing error for those customers without Variance Parameter meters (because their meter already measures total consumption).

Use **SQ Rule Processing** to indicate if the SQ rule is `Always` executed by rate application, even if the consumption details on the original bill segment are to be used when the system recalculates a bill segment (refer to Rebill (Bill Segment)). If the SQ rule should not be executed when the consumption details on the original bill segment are used, indicate that the SQ rule should only be executed on the `Initial SQ Calculation`. SQ rules that create a characteristic type / value (in the characteristic collection) used by one or more bill factors referenced in a rate component should typically be always executed. Refer to Base Package SQ Rules for examples of SQ rules that create a characteristic type / value.

Select the **SQ Algorithm Type** that controls how the system manipulates the consumption supplied to the rate. Define the **Parameters** to be used by the SQ algorithm.

Base Package SQ Rules

Note: If the algorithms described below are not sufficient for your rating requirements, you must write a new SQ rule algorithm. Refer to How To Add A New SQ Rule for how to do this.

The following table lists and describes the SQ rule algorithms shipped with the product:

SQ Algorithm	Alg Type	What It Does	Parameters Used
`Bill days (Days)`	DY	This SQ rule calculates the total number of days in the bill period. This algorithm is necessary if you have charges that are based on the number of days of service (e.g., a minimum charge that varies based on the days of service). Refer to Calculated Minimum Charges and Step Boundaries Based On The Number Of Days Of Usage And The Region In Which The Customer Resides for good examples of when to use such an SQ rule.	No parameters are used.
`Characteristic Value`	CV	This SQ rule creates an entry in the SQ array containing the sum of the characteristic values of a given Characteristic Type. The values are accumulated from the entity defined in the parameter list. Resulting Quantity (expressed in the Output UOM / Sub-	Param 1 = Characteristic Type Param 2 = Characteristic Source Param 3 = Minimum Value

Copyright © 2024, Oracle and/or its affiliates.

SQ Algorithm	Alg Type	What It Does	Parameters Used
		UOM / SQI) = the sum of the characteristic values of a given **Characteristic Type**. **Characteristic Source** determines the entity that holds the characteristic values: • If **Characteristic Source** is "contract", the characteristic value effective on the bill start date for the Characteristic Type (P1) and contract ID is returned. • If **Characteristic Source** is "ACCOUNT", the characteristic value effective on the bill start date for the Characteristic Type (P1) and Account Id is returned. • If **Characteristic Source** is "LOCATION", the characteristic values are accumulated from all locations linked to the contract ID. • If **Characteristic Source** is "SP", the characteristic values are accumulated from all service points linked to the contract ID.	
		If **Minimum Value** is greater than zero and the resultant characteristic value is less than the specified **Minimum Value**, an error is generated.	
`Contract Quantity`	CQ	This SQ rule is used to create an entry in the SQ array for a given Contract Quantity Type specified in a contract. If multiple values are effective during a bill period, the Proration Rule determines what happens. (See below). Resulting Quantity (expressed in the Output UOM / Sub-UOM / SQI) = the value of a given **Contract Quantity Type** in a contract. If multiple values are effective during a	Param 1 = Contract Quantity Type Param 2 = Proration Rule: CQBD, CQED, CQMA, CQMI, CQPR

Copyright © 2024, Oracle and/or its affiliates.

SQ Algorithm	Alg Type	What It Does	Parameters Used
		bill period the **Proration Rule** determines what happens: • If **Proration Rule** is 'Use Contract Quantity effective at Beginning of Bill Period' (i.e., `CQBD`), SQ Amount = Contract Quantity effective at the start of the bill period. • If **Proration Rule** is 'Use Contract Quantity effective at End of Bill Period' (i.e., `CQED`), SQ Amount = Contract Quantity effective at the end of the bill period. • If **Proration Rule** is 'Maximum' (i.e., `CQMA`), SQ Amount = largest Contract Quantity effective in the bill period.	
		• If **Proration Rule** is 'Minimum' (i.e., `CQMI`), SQ Amount = smallest Contract Quantity effective in the bill period • If **Proration Rule** is 'Prorate' (i.e., `CQPR`), SQ Amount = Sum of all prorated contract quantities (calculate the number of days in the bill period (X); calculate the number of days in the bill period that each contract quantity is effective (Y(i)); prorate each contract quantity (i) = (Contract Quantity(i) * Y(i) /X)).	
`Direct Access Credits`	CD	This SQ rule creates a characteristic type / value to handle credits issued to the customer for participating in direct access. These values are used by one or more bill factors in the rate components to vary the amount associated with a given bill line.	
`Early Pay`	EP	This SQ rule is only used if you provide customers with early payment discounts via	Param 1 = Char Type - Number of Days Early

Copyright © 2024, Oracle and/or its affiliates.

SQ Algorithm	Alg Type	What It Does	Parameters Used
		a discount on their next bill segment. This rule works as follows: • First, it determines if the prior bill segment's bill was paid before its due date. It does this by retrieving the customer's balance as of the prior bill's due date - 1 day. If this amount is <= zero, it checks if a payment exists with a payment date on or after the bill's bill date and before the bill's due date. If yes, processing continues. • A characteristic type / value is created where the value holds the number of days early (and the characteristic type is equal to **Char Type - Number of Days Early**).	Param 2 = Char Type - Eligible For Early Pay Disc Param 3 = Char Value - Eligible For Early Pay Disc
		• In addition, this routine accumulates the sum of the bill lines on the prior bill segment for which the early payment discount is eligible. It does this by accumulating all bill lines that reference a given characteristic type and value (where characteristic type is equal to **Char Type - Eligible For Early Pay Disc**, and **Char Value - Eligible For Early Pay Disc**). It populates the total amount in the service quantity defined on the SQ rule. This means that you must define this characteristic type / value on each rate component whose resultant value is eligible to receive an early payment discount. This rules anticipates the early pay discount being levied using two rate components:	
		• The first would be an FCPO rate component that holds the amount eligible	

Copyright © 2024, Oracle and/or its affiliates.

SQ Algorithm	Alg Type	What It Does	Parameters Used
		to receive the discount. This would be an SQ rate component where the service quantity is the value produced by this SQ rule, and the price would be 1 (or -1). In addition, this rate component would need eligibility rules so that it is only applied if the number of days early (held in the characteristic type / value produced by this algorithm) is > 0. • The second would be an "apply to" rate component (where the apply to value is a percentage applied to the first rate component). In addition, this rate component would need eligibility rules so that it is only applied if the number of days early is within a certain tolerance (e.g., > 0). You could create several of these types of rate components if the discount percent differs based on the number of days early.	
`Math`	MA	This SQ rule performs mathematical functions on service quantities prior to rate application. There are three major groups of parameters: 1) **Quantity 1** (identified by a service quantity's UOM/Sub-UOM/SQI). 2) **Mathematical operand** 3) **Quantity 2** (identified by a service quantity's UOM/Sub-UOM/SQI or a Bill Factor) For example, if you want to add together two SQ entries, these values would contain: 1) the unique identifier of one of the SQ's (in parameters 1 through 3)	Quantity 1: Param 1 = UOM of service quantity entry Param 2 = Sub-UOM of service quantity entry Param 3 = SQI of service quantity entry Or Param 9 = Decimal number Mathematical Operand: Param 4 = operand symbol (i.e., +, -, *, /) Quantity 2: Param 5 = UOM of service quantity entry Param 6 = Sub-UOM of service quantity entry

Copyright © 2024, Oracle and/or its affiliates.

SQ Algorithm	Alg Type	What It Does	Parameters Used
		2) the "+" symbol (in parameter 4) 3) the unique identifier of the other SQ (in parameters 5 through 7) It should be stressed that **Quantity 2** can either reference a **bill factor** (param 8) or an SQ entry (parameters 5 through 7). If both are specified, the bill factor is used. The two **default values** (param 9 and param 10) are only used if Quantity 1 and/or Quantity 2 reference SQ's / bill factor that are not present at billing time. There are three ways to use these **default values**:	Param 7 = SQI of service quantity entry Or Param 8 = Bill factor ID Or Param 10 = Decimal number
		• If you want to multiply a SQ by a constant value, define the SQ in Quantity 1 and the constant value in Param 10, i.e., you don't have to specify anything in Param's 5 through 8. • If you want a default value used if the system cannot find a SQ that matches the UOM/Sub-UOM/SQI, define this value in Param 9 and/or Param 10. • If you use a bill factor to define Quantity 2 (in Param 8) and you want a default value used if the system cannot find a char type / value for a specific customer, the default value is defined in Param 10. The following **mathematical operands** use only **Quantity 1**. ABS, NEGATE, ROUND, FLOOR (meaning to round down), CEILING (meaning to round up), SIN, ASIN, COS, ACOS, TAN, ATAN, LOG, LOG10, EXP, EXP10, SQRT	

Copyright © 2024, Oracle and/or its affiliates.

SQ Algorithm	Alg Type	What It Does	Parameters Used
		The following **mathematical operands** use both **Quantity 1** and **Quantity 2**: +, -, *, /, **, MOD, <, <=, >, >=, =, <>, MAX, MIN, AVG The following **mathematical operands** use both **Quantity 1** and **Quantity 2** and return a value of 1 if the comparison is true and a value of 0 if the comparison is false: <, <=, >, >=, =, <>	
`Maximum Service Quantity`	MQ	This SQ rule compares a quantity (either measured or created via a previous SQ rule) against a contract quantity and returns whichever is larger. Resulting Quantity (expressed in the Output UOM / Sub-UOM / SQI) = greater of quantity for **UOM/ Sub-UOM/ SQI** OR contract quantity for **Contract Quantity Type**	Param 1 = UOM Param 2 = Sub-UOM Param 3 = SQI Param 4 = Contract Quantity Type
`Required UOM/TOU Validation`	RP	This SQ rule verifies if one UOM/Sub-UOM quantity is non-zero then the other should be non-zero too. **UOM1/Sub-UOM1/SQI1** is the ID of the first SQ entry. **UOM2/Sub-UOM2/SQI2** is the ID of the second SQ entry.	Service Quantity ID1 • Param 1 = UOM1 • Param 2 = Sub-UOM1 • Param 3 = SQI1 Service Quantity ID2 • Param 4 = UOM2 • Param 5 = Sub-UOM2 • Param 6 = SQI2
`Sum`	SM	This SQ rule adds up several quantities (either measured or created via a previous SQ rule). This algorithm would be useful when you need to add together "Sub-Unit of Measure" registers because there are charges in the rate that are for the total amount	Quantity 1 Param 1 = First UOM Param 2 = First Sub-UOM Param 3 = First SQI Quantity 2 Param 4 = Second UOM

Copyright © 2024, Oracle and/or its affiliates.

SQ Algorithm	Alg Type	What It Does	Parameters Used
		of consumption regardless of when it was used. Resulting Quantity (expressed in the Output UOM / Sub-UOM / SQI) = Quantity for **UOM** (in param 1) **Sub-UOM** (in param 2) **SQI** (in param 3) + Quantity for **UOM** (in param 4) **Sub-UOM** (in param 5) **SQI** (in param 6) + Quantity for **UOM** (in param 7) **Sub-UOM** (in param 8) **SQI** (in param 9)	Param 5 = Second Sub-UOM Param 6 = Second SQI Quantity 3 Param 7 = Third UOM Param 8 = Third Sub-UOM Param 9 = Third SQI

For more information about defining algorithm types, refer to *Setting Up Algorithm Types*.

Where Used

A Rate Schedule may have zero or more SQ Rules. Refer to Rate Schedule - SQ Rules for more information.

Defining Service Quantity (SQ) Identifiers

CAUTION: SQIs exist to support calculations performed by rate components. This means that you must design your rate components before you can design your service quantity identifiers. We strongly recommend that you design "on paper" how every rate's rate component looks before you attempt to set up your SQIs.

Note: If you create an SQI, you must also have a corresponding SQ Rule to calculate the amount of the SQI to be priced by the rate. If not, it is impossible for the system to generate a bill line for the SQI because there will be no quantity associated with the SQI.

To define service quantity identifiers, open **Admin Menu, Service Quantity Identifier**.

Description of Page

Enter a unique **Service Qty. Identifier** and **Description** for every SQI.

Enter the appropriate number of **Decimal Positions** for each SQI. The rate application process uses this information to round the calculated service quantity prior to applying appropriate rate components.

Where Used

An SQ Rule may use an SQ Identifier to describe what it generates. Refer to Setting Up Service Quantity (SQ) Rules for more information.

A Service Quantity Rate Component may use an SQ Identifier to describe what it is pricing. In this situation, the rate must contain an SQ Rule that generates a quantity for this SQI; otherwise there will be a price without a service quantity. Refer to Rate Component - Main Information for more information.

Setting Up A Rate Schedule

A rate schedule is setup for every rate that you offer to your customers. Rate schedules are defined using the pages described in this section.

Copyright © 2024, Oracle and/or its affiliates.

> **Note: A rate schedule isn't complete without rate versions and rate components.** Refer to All Rates Share A Common Structure for information about how you must also create rate versions and rate components for every rate schedule.

Rate Schedule - Main

You start the rate definition process by selecting **Main Menu, Rates, Rate Schedule**. On this page you enter general information about the rate.

Description of Page

Enter a unique **Rate Schedule** ID to identify the rate. The value that you supply will typically correspond with the identity of the rate in your rate book.

Enter a **Description** for the rate schedule.

Select the **Service Type** that describes the service associated with this rate. When you maintain an existing rate schedule, this field will be gray if the rate is referenced on at least one contract type. See *Setting Up Service Types* for more information.

Select a **Frequency** to define the time period in which the rate's charges are expressed. See Defining Frequency Codes for more information.

Select a **Currency Code** to define the currency in which the rate's prices are expressed. See *Defining Currency Codes* for more information.

> **Note: Currency note.** All bill factors that are referenced on the rate's rate components must be denominated in the same currency as defined on the rate schedule. This is because bill factors are used when the price is "soft" and all prices in a rate, be they hard or soft, must be in the same currency. Also note - all rates referenced on an account's contracts must be denominated in the same currency as defined on the account.

Turn on **Allow Estimates** if the bill process is allowed to estimate consumption for contracts that use this rate. Consumption is estimated if all of the following points are true:

- A real read cannot be found.
- The contract allows estimation.
- The bill cycle schedule allows estimation (if the bill is produced by the BILLING background process).
- The rate schedule allows estimation.

Turn on **Allow RVs Proration** if a change in the Rate Version during a bill period should be prorated. If changes should not be prorated, indicate the **RV Selection Date** to use when the system determines which rate version to use.

The following table lists and describes the valid rate version selection date options:

Option	Description
`Accounting Date`	The rate version effective on the bill's accounting date will be used.
`Bill End Date`	The rate version effective on the bill's end date will be used.
`Bill Start Date`	The rate version effective on the bill's start date will be used.

Rate Schedule Tree

The right half of this page is dedicated to a tree that shows the rate schedule's SQ Rules, Register Rules and Rate Versions. You can use this tree to view high-level information about these objects. You may also transfer to a given rate version by selecting that rate version.

> **Note:** The duplicate action in the action button bar enables you to copy another rate schedule and optionally, its rate versions. Refer to Duplicate Button in the system wide standards document for more information.

Copyright © 2024, Oracle and/or its affiliates.

Where Used

Most contract types contain a collection of valid Rate Schedules. Refer to *Contract Type - Rate* for more information.

Most contracts reference a Rate Schedule. Refer to Contract - Rate Info for more information.

Rate Schedule - SQ Rules

When a rate schedule has SQ Rules, the system executes each rule's algorithm before it calculates the charges embodied in the rate's rate components. These SQ Rules manipulate measured consumption so that the rate is supplied with consumption in the manner that it expects. Open **Rates, Rate Schedule** and navigate to the **SQ Rule** to link SQ Rules to a rate.

Fastpath: For more information **,** refer to Setting Up Service Quantity (SQ) Rules.

Note: The Rate Schedule Merge page enables you to build the SQ rule collection for your rate schedule by copying collections from existing rate schedules.

Description of Page

Enter a **Service Quantity Rule** for every type of service quantity manipulation required by the rate. The **Sequence** number is important when you have multiple rules linked to the rate, as it defines the order in which the rules will be executed.

Where Used

Billing uses this information to create additional service quantities at billing time.

Rate Schedule - Register Rules

When a rate schedule has register rules, the system executes each rule's algorithm before it calculates charges. These register rules convert measured consumption so that the rate is supplied with consumption in the manner that it expects.

Fastpath: For more information about register rules, refer to Setting Up Register Rules.

Note: The Rate Schedule Merge page enables you to build the register rule collection for your rate schedule by copying collections from existing rate schedules.

Select **Main Menu, Rates, Rate Schedule** and navigate to the **Register Rule** to link register rules to a rate.

Description of Page

Enter a **Register Rule** for every type of register rule to be executed before consumption is rated. The **Sequence number** is important when you have multiple rules linked to the rate, as it defines the order in which the rules will be executed.

Where Used

Billing uses this information to create or adjust service quantities at billing time.

Rate Schedule - Bill Messages

When a rate schedule has bill messages, the system will sweep these messages onto bills that use the rate.

Fastpath: For more information about bill messages, refer to The Source Of Bill Messages.

Note: The Rate Schedule Merge page enables you to build the bill message collection for your rate schedule by copying collections from existing rate schedules.

Copyright © 2024, Oracle and/or its affiliates.

Select **Main Menu, Rates, Rate Schedule** and navigate to the **Bill Messages** to link bill messages to a rate.

Description of Page

Use the **Bill Messages** collection to define **Bill Message** codes that should appear on bills that use a given rate schedule. For each message, also specify the **Start Date** and **End Date** when such a message should appear on the bill (leave **End Date** blank if the message should appear indefinitely).

Where Used

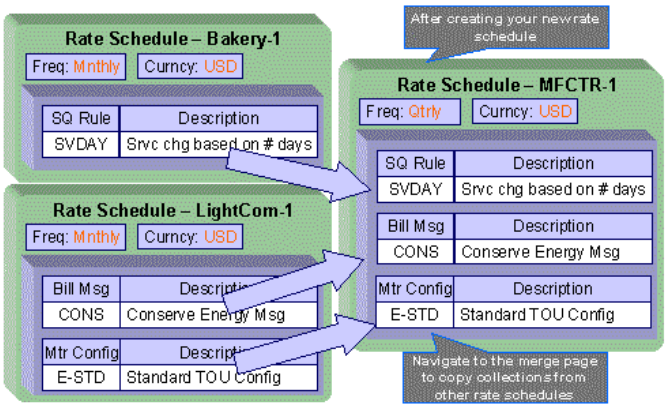
The system snaps bill messages on a bill during bill completion. Refer to The Source Of Bill Messages for more information.

Rate Schedule Merge

Use this page to modify an existing rate schedule by copying information from other rate schedules. This page may be used to copy records from the SQ rule, register rule, and bill message from one or more existing rate schedules to another. Some examples of when this page may be used are as follows:

- You wish to create a new rate schedule, which is similar to an existing rate schedule. Rather than copying all the information from the existing rate schedule and then removing the inapplicable components, this page may be used to selectively copy only the information applicable to the new rate schedule.
- Perhaps you offer a rate to your customer, which contains many optional components. Rather than building a generic rate schedule used by many customers, and using other logic to determine which options are applicable to which customer, you may choose to build a custom rate schedule for each customer, using only the components applicable to that customer. In this scenario, you may choose to create special 'mini' rate schedules, one for each of the various options. Then, you could use the rate schedule merge page to select the components applicable for the new custom rate schedule.
- Perhaps you are adding several new bill messages, which are applicable to multiple existing rate schedules. Once you have added the new messages to one rate schedule, you may find it easier to update the subsequent rate schedules by using the rate merge page to copy the messages across.

Note: The target rate schedule must exist prior to using this page. If you are creating a new rate, you must first go to the Rate Schedule page to add the new rate schedule and then navigate to the merge page to copy collection information.



Note: Duplicate versus Merge. The Rate Schedule page has Duplication capability. You would duplicate a rate schedule if you want to a) create a new rate schedule AND b) populate it with *all* the information from an existing rate schedule. You would use the rate schedule merge page if you want to build a rate schedule using pieces of one or more rate schedules.

Open **Main Menu, Rates, Rate Schedule Merge** to open this page.

Description of Page

Copyright © 2024, Oracle and/or its affiliates.

Select the **Original Rate Schedule** that is the target for merging the rate schedule collection information.

Select the **Merge From Rate Schedule** that is your template rate schedule to copy the collections from.

> **Note:** You may only copy information from one Merge From rate schedule at a time. If you wish to copy information from more than one rate schedule, select the first Merge From rate schedule, copy the desired records, Save, then select the next Merge From rate schedule.

The left portion of the page will display any existing records in the collections for the original rate schedule. The right portion of the page will display the existing records in the collections for the Merge From rate schedule.

You may use the **Copy All** button to copy all the records in all the collections from the Merge From rate to the Original rate. If you do not choose to copy all, you may copy records individually as described below.

The left portion of the **SQ Rule** collection initially displays existing SQ rule records linked to the original rate schedule. In the **Merge Type**, you will see the word `Original`, for any of these records. The **Sequence** number and **SQ Rule** description are displayed. In the right portion of the collection, the existing records in the merge from rate are displayed initially.

The left portion of the **Register Rule** collection initially displays existing register rule records linked to the original rate schedule. In the **Merge Type**, you will see the word `Original`, for any of these records. The **Sequence** number and **Register Rule** description are displayed. In the right portion of the collection, the existing records in the merge from rate are displayed initially.

The left portion of the **Bill Messages** collection initially displays existing bill messages linked to the original rate schedule. In the **Merge Type**, you will see the word `Original`, for any of these records. The description of each **Bill Message** is displayed. In the right portion of the collection, the existing records in the merge from rate are displayed initially.

The topics, which follow, describe how to perform common maintenance tasks:

Removing A Row

If you wish to remove a record linked to the Original rate schedule, press the "-" button to the left of the record.

Adding A New Row

You may move any of the records from the Merge From rate to the original rate by selecting the left arrow adjacent to the desired row. Once a record is moved it will disappear from the Merge From information and appear in the Original information with the word `Merge` in the Merge Type column.

Removing An Uncommitted Row

If you have copied a row across by mistake, you may remove it by clicking on the right arrow adjacent to the appropriate record.

Moving Rows Up and Down

The **SQ Rule** and **Register Rule** grids contain sequence numbers that indicate the order in which the rules should be executed. You may modify the execution order using the up and down arrows.

You may also need to renumber the sequences so that your rules are executed in the proper order. The fields **Start from sequence number** and **Increment By** are used to define the sequence number to assign to the first rule and the subsequent values to assign to subsequent rules.

> **Note:** **Note** that once the record is saved, both register rules are marked as `Original`.

Copyright © 2024, Oracle and/or its affiliates.

Fastpath: Refer to Editable Grid in the system wide standards documentation for more information about adding records to a collection by selecting from a list and repositioning rows within a grid.

Setting Up Bill Factors

This section describes bill factors and how they are used to levy the many charges referenced in a rate.

CAUTION: Most bill factors exist to support calculations performed by rate components. This means that you must design your rate components before you can design your bill factors. We strongly recommend that you design "on paper" how every rate's rate components looks before you attempt to set up bill factors.

An Overview Of Bill Factors

The primary reason why rate components exist is to define how much to charge for the services sold by your company. When you create rate components, you have four ways to indicate how much to charge:

- Specify the value directly in the rate component.
- Tell the rate component to use the value defined in a *bill factor*.
- Tell the rate component to use the value calculated by a " *for calculation purposes only*" rate component.
- Tell the rate component to call an algorithm to get the price.

The remainder of this section describes the bill factor approach. The other methods are discussed in Designing Rate Components and in Rate Examples.

Bill factors are frequently used to specify a tax rate. For example, you could specify the current tax rate (say 6%) directly in every rate's rate components OR you could indicate every rate should levy the current state tax rate and have the system look up the applicable rate when it calculates bills. The latter approach involves using a bill factor to indicate that the amount to charge should be retrieved from someplace other than the rate component when a bill is calculated.

You can use bill factors to define any of the following types of charges:

- A flat amount. This would typically be used to define service charges, minimum charges, and asset charges.
- An amount for a given unit of measure. This would typically be used to define the price for product/charge type consumption (e.g., the price for the product).
- A percentage. This would typically be used for tax rates and discount percents.

As a rule of thumb, you would use a bill factor (rather than specify the value in the rate component) when any of the following situations exist:

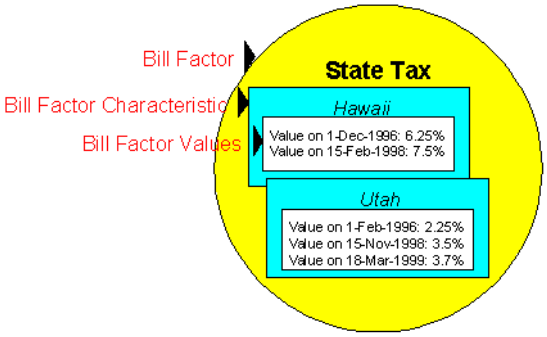
- The same charge exists in many rates. For example, if there are 3 rates that all contain the same service charge, it would make sense to use a bill factor to levy this charge rather than specify the same value on 3 rate components.
- The amount being charged is dictated by some external organization and therefore can change independently from the rate. For example, tax percentages and use fees are set by taxing authorities and can be changed at any time (and you don't want to change the rate when these charges change).
- The amount being charged varies depending on WHERE the customer lives. If you have a rate where the price per gallon differs depending on where the customer lives, you should use a bill factor to levy this type of charge.
- A charge (or discount) is only levied for a subset of customers for defined periods of time. For example, if you have a discount that is only given to customers who have had service for more than 2 years, you would use a bill factor to specify the discount percent.
- The price differs per customer. For example, you may have a rate for a customer where the price of product is different for every customer. In this situation, you would use a bill factor to specify the price.
- The price differs based on the number of weeks of consumption. Where the price charge is based on the weighted average spot market price in effect during the bill period. Refer to Interval Prices Can Be Used For More Than Interval Prices for more information.

Copyright © 2024, Oracle and/or its affiliates.

> **CAUTION:** As you can deduce, bill factors are extremely flexible. While the flexibility makes for a very powerful rating engine, it also presents you a dizzying array of options. We have discovered that to take advantage of the flexibility, you need to gain an overall understanding of the pricing and taxation provisions in ALL of your rates. Only then can you make logical and pragmatic decisions in respect of the number and type of bill factors.

The Structure Of A Bill Factor

All bill factors share a common structure as illustrated below:



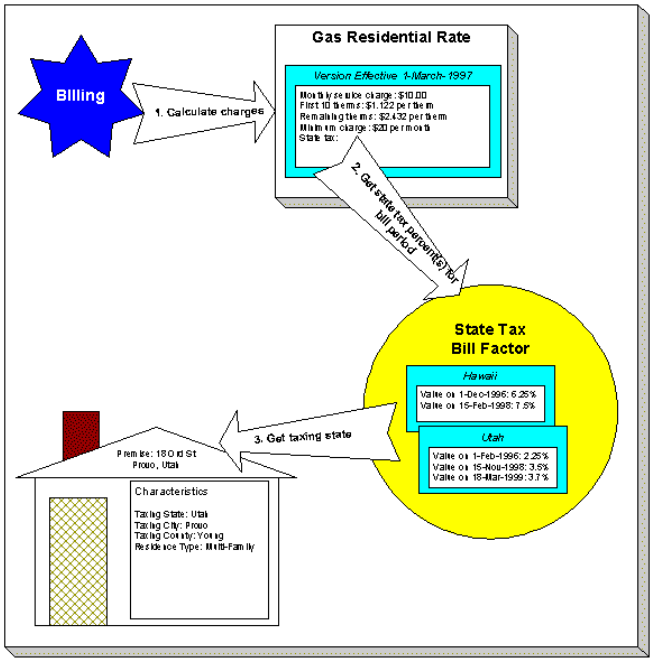
The above example shows a standard bill factor made up of the following tables:

- A **bill factor** contains descriptive information and attributes that control how the bill factor may be used in the system.
- All bill factors have the capability of having different values depending on some characteristic. For example, a bill factor used to levy a state tax would have a different tax rate depending on the state in which the customer resides. A **bill factor characteristic** must exist for every state with a tax (states without a tax will not be linked to the state tax bill factor). At billing time, if the system cannot find a bill factor characteristic that corresponds with the customer's state of residence, the charge will be skipped.
- And finally, each bill factor characteristic may have many **bill factor values** over time. To continue with our state tax example, each state with a tax can have different tax rates over time. There are different ways in which a bill factor's values may be defined:
  - For bill factors whose values are applicable to many customers where only one value is effective on a given date, e.g., a city tax, the values are defined using bill factor values. Refer to Defining Bill Factor Values for more information.
  - For bill factors whose values are applicable to many customers where more than one value is effective on a given date, e.g., an interval price or a weighted average price, the values are defined on an interval value set. Refer to Defining Interval Values for more information about interval value sets.
  - For bill factors that define values for a collection of Variance Parameter codes for a given date, the values are defined using bill factor Variance Parameter values. Refer to Bill Factor Variance Parameter Values for more information about Variance Parameter.
  - For bill factors whose value differs per customer, the value is defined in the customer's contract. Refer to Contract - Contract Value for more information about bill factors in contract terms.

An Illustration Of A Bill Factor And Its Characteristics

The following picture illustrates how a bill factor and its characteristics are used to retrieve the relevant state tax at billing time:

Copyright © 2024, Oracle and/or its affiliates.



The following points describe the above:

- At billing time, the billing engine sends a request to a rate to calculate the charges for a customer's consumption.
- The rate calculates charges without bill factors until it encounters the rate component used to levy state tax. This rate component references a bill factor. This means the rate must get the tax rate(s) in effect during the billing period from this bill factor.
- The state tax bill factor contains an attribute defining taxing state characteristic controls the bill factor value. The bill factor therefore extracts the taxing state.

Note: Deriving characteristic values. Rather than have the system extract the characteristic value from an entity, you can setup the system to derive the characteristic value when the rate is calculated. For example, if all of your customers are located in a single state, you may not want to maintain the taxing state on every location. To do this, you could setup the rate to "hard code" a taxing state of say Hawaii. This is an advanced topic, but it may prove useful for your implementation. Refer to Deriving / Passing In Characteristic Values for the details.

- The taxing state and the bill factor extracts the tax rate(s) in effect during the bill period and returns them to the rate.
- The rate applies the tax percents and returns the charges to billing.

Note: Some bill factors don't need a characteristic. There are bill factors whose value does not differ based on a characteristic. every bill factor value must reference both a bill factor and a characteristic value. Therefore, if you have bill factors whose value is not related to a characteristic, you must specify a characteristic type on the bill factor with a characteristic value that is exactly the same as the characteristic type.

Fastpath: For more information about setting up characteristics, see *Setting Up Characteristic Types & Their Values*.

Deriving / Passing In Characteristic Values

CAUTION: This is an advanced, technical concept.

This is a very common technique and is easy to understand. However, there are situations when an entity in Oracle Revenue Management and Billing doesn't contain the appropriate characteristic value and therefore other techniques must be used. These techniques may not apply to your implementation, but they're worth understanding to help you form an intuitive understanding of rate application:

Copyright © 2024, Oracle and/or its affiliates.

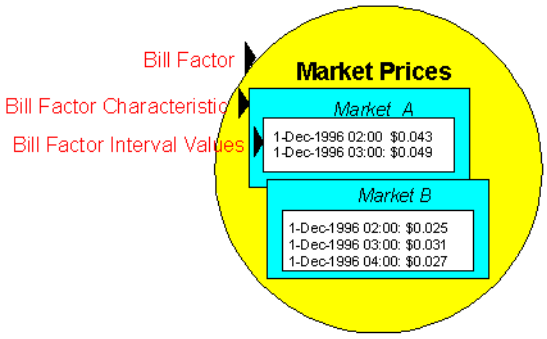
- There are implementations that only use the Oracle Revenue Management and Billing rating engine. These implementations do not setup accounts, etc. Rather, they simply setup their rate schedules, rate components, and bill factors and then call the rating engine from another application. This means the rate engine can't retrieve characteristic values from within Oracle Revenue Management and Billing (because there are no objects on which the characteristic values can be defined). In this situation, the application that calls the rate engine must supply the characteristic types and values to be used when a bill factor requires this type of information.
- Assume you have a charge that is based on the account's customer class and the contract's revenue class. This type of charge's characteristic value is not extractable from a single entity. Rather, the system must construct the characteristic value real-time by concatenating information from the account and contract being billed. In this situation, you can use an SQ rule to derive this characteristic type and value. This SQ rule would create a characteristic type and value real time by concatenating the account's customer class and the contract's revenue class. This characteristic type and value only exists in memory while rate application executes. This is an example of the system deriving a characteristic value as opposed to it being passed in or retrieved from another object in the system.

When you setup a bill factor to use a characteristic whose value is derived or passed in, you need to define its characteristic source as `Characteristics Collection`. This source tells the system to extract the characteristic value from memory rather than extracting it from an object within the system.

Note: The various plug-in spots on a rate component have access to the `Characteristics Collection`. You might find this useful if you have a rate component that needs to calculate characteristic values for subsequent rate components.

Bill Factor Interval Values

A bill factor may also contain interval values.



Interval values are typically used to record interval prices (i.e., prices that are applicable to short time intervals of say 15 minutes). Just as with simple bill factors, they may have characteristics defined. In the above example, the characteristic is the "Market" where the prices are defined. This bill factor probably indicates that the contract has a characteristic indicating the market from which it gets its prices. This enables you to use a common rate for customers who get their prices from different markets.

Interval Values Can Be Used For More Than Just Interval Prices

The interval value transaction can be used for more than recording interval prices. One such purpose is to record rolling average prices. Let's use an example to make the point. Assume the following:

- You operate in a deregulated environment and part of the price charged to your customers is based on the spot-market price.
- Your customers don't have interval meters. Rather, they simply have meters that are read once a month. When you calculate a customer's bill, you must determine the single price to be applied to the total amount of consumption for the month.
- The price applied to a customer's bill is based on the number of weeks of consumption and the end date of the bill period. For example, a customer whose bill ends on 12-Sept-2001 with 3 weeks of consumption would probably

Copyright © 2024, Oracle and/or its affiliates.

have a different price than a customer whose bill ends on the same date but has 6 weeks of consumption. Why? Because the spot market price varies during the different billing periods.

- Different prices may be calculated based on the customer's load profile. This would mean that some customer whose bill ends on 12-Sept-2001 with 3 weeks of consumption would have a different price than some other customer whose 3 week bill ends on the same date. Why? Because some customers tend to consume in different time periods than other customers (and therefore different spot market prices were in effect).

If you assume the above, then you'd need the ability to record different prices based on:

- The bill end date
- The number of weeks / days in the billing period
- The customer's load profile

Below, we've illustrated how interval pricing information might look for a bill factor that holds the weighted average prices.

Interval Data is Stored in Data Sets

Interval data in the system will have many different origins.

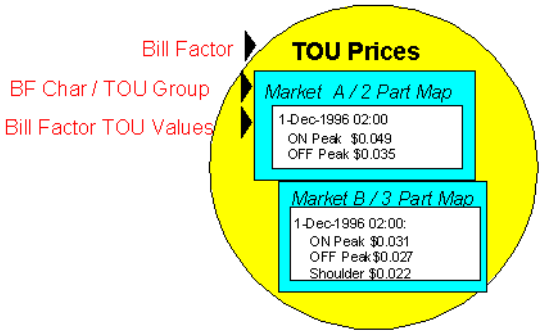
- It may be contract-related data, which is setup at the beginning of the contract or contract year
- It may be data, which is interfaced from an external source periodically

Regardless of the origin of the data, all data will have similar characteristics

- Data will be received by the system, or created by the system periodically. The system will store whatever new data needs to be stored.
- The period represented by the batch of new data is variable.
- New data may be an update to previously received data

Bill Factor Variance Parameter Values

A bill factor may also contain values, which differ by a variance parameter. This enables you to set up prices that differ by variance parameter or to set up contract values, which differ by variance parameter.



Bill Factor Variance Parameter values are typically used to record prices, which vary by variance parameter code. Variance Parameter prices are used in conjunction with variance parameter mapping. The customer would typically have interval quantities, which are mapped, using a Variance Parameter Map, into a collection of variance parameter quantities. Defining a bill factor with prices per Variance Parameter allows you to set up a single rate component to apply charges to these quantities, resulting in multiple bill calculation lines.

Just as with simple bill factors, Variance Parameter bill factors may have characteristics defined. In the above example, the characteristic is the "Market" where the prices are defined. This bill factor probably indicates that the contract has a characteristic indicating the market from which it gets its prices. This enables you to use a common rate for customers who get their prices from different markets.

Copyright © 2024, Oracle and/or its affiliates.

In addition to defining the characteristics, Variance Parameter bill factors must also indicate the Variance Parameter Group. The Variance Parameter group is a collection of time period definitions, which make up a Variance Parameter map. This enables you to define different prices for On Peak, when it is part of a 2 part Variance Parameter map versus when it is part of a 3 part Variance Parameter map.

Fastpath: Refer to *Grouping of Variance Parameter Codes* for more information.

The Variance Parameter prices are defined for the combination of characteristic and Variance Parameter Group. In the examples below, a bill factor's characteristic is State. Also note that the effective date of the prices are the same for all variance parameter codes. If the price of any Variance Parameter value changes, then a new set of prices for all Variance Parameter codes must be entered.

TOUs in Group 2	California
	Effective 1 Jul 1999
On Peak/Summer	$0.98
Off Peak/Summer	$0.74
On Peak/Winter	$0.84
Off Peak/Winter	$0.63

TOUs in Group 3	California
	Effective 1 Jul 2000
On Peak/Spring	$1.03
Off Peak/Spring	$0.74
On Peak/Autumn	$0.92
Off Peak/Autumn	$0.63

TOUs in Group 2	Oregon
	Effective 1 Jul 1999
On Peak/Summer	$0.98
Off Peak/Summer	$0.74
On Peak/Winter	$0.84
Off Peak/Winter	$0.63

The application of these variance parameter values requires the use of an algorithm. If your business uses the interval billing subsystem and uses variance parameter mapping and pricing, then functionality is available for applying these values to interval quantities, which have been mapped into variance parameter quantities. Refer to *The Big Picture of Interval Billing* for more information.

Variance Parameter Values Can Be Used For More Than Just Prices

Variance Parameter values can be used for more than recording prices. Many contract-based values may also be recorded per Variance Parameter. For example, a contract for an interval billing customer may designate a hedge cover or a minimum demand value or a contract demand value. Although you may have interval quantity curves to define these values, that would be difficult to define and maintain. A more common scenario would be to define a Variance Parameter Map and to define these values for each Variance Parameter. A Variance Parameter Bill Factor could be used to define these values.

Variance Parameter Values May Differ By Customer

In the same way that simple bill factors may be used to define customer specific prices and values, Variance Parameter bill factors may also be used to define customer specific prices and values.

Defining Bill Factors

You create a bill factor for every type of charge whose value is not specified directly in a rate component.

You use two windows to set up a bill factor. On the first, you define general information describing how the bill factor is used in the system. On the second, you define every characteristic value for which the bill factor has a value.

Defining General Bill Factor Information

You begin to define a bill factor by selecting **Main Menu, Rates, Bill Factor**.

Description of Page

Copyright © 2024, Oracle and/or its affiliates.

Enter a unique **Bill Factor** code and a **Description** for the bill factor.

Use **Bill Factor Type** to indicate if this bill factor uses `Regular`, `Interval` (i.e., real time) or `Variance Parameter` values. If you choose `Regular`, this bill factor's values are specified on Bill Factor Value - Main or on Contract - Contract Values . If you choose `Interval`, this bill factor's values are specified on Defining Interval Values. If you choose `Variance Parameter`, this bill factor's values are specified on Bill Factor Variance Parameter Values.

Enter a **Currency Code** to define the currency in which the bill factor's prices are denominated. This field will be gray if the bill factor is referenced on a rate component.

Note: Bill factors referenced on a rate component must be denominated in the same currency as is the rate.

Select the **Value Type** to define the kind of value maintained by this bill factor.

The following table lists and describes the valid value types for a bill factor:

Value Type	Description
`Charge`	The value maintained in this bill factor is a fixed amount (e.g., a flat charge).
`Percentage`	The value maintained in this bill factor is a percentage.
`Unit Rate`	The value maintained in the bill factor is a price per service quantity.

This field will be gray if the bill factor is already referenced on a rate component or if the bill factor already has values.

Turn on **Error If No Value** if the system should create a billing error when a bill factor value doesn't exist for the characteristic value linked to the customer. Confused? To clarify this, you need to be aware that there are different types of bill factors:

- Those that should have a value for every customer whose rate references the bill factor. A bill factor used to levy the relevant state tax is a good example (assuming every state has a tax). For this type of bill factor, if the system cannot find a bill factor value for the taxing state, it must generate a billing error because data is missing from the system. You would turn on **Error if No Value** for this type of bill factor.
- Those that only have a value when the customer has a characteristic value that levies a charge. A bill factor used to levy the relevant city tax is a good example (because not all cities levy taxes). For this type of bill factor, if the system cannot find a bill factor value for the city in which the customer lives, the rate component will be skipped and no billing error will be generated. You would turn off **Error if No Value** for this type of bill factor.

Bill factors used to levy taxes may or may not be eligible for tax exemptions; it's up to the respective taxing authority. Customers who are eligible for the tax exemption will have their tax obligation reduced (or nullified) at billing time. For bill factors that are eligible for tax exemptions, turn on **Tax Exemption**. If a contract is fully or partially exempt for the taxes defined by a given bill factor, the bill factor and the exemption information for the customer must be indicated on the Contract page. This field will be gray if the bill factor is referenced as being tax exempt in a contract.

Note: Tax Exemptions limited to Apply To Rate Components. Although bill factors marked as eligible for tax exemptions may be referenced on several different types of rate components, only Apply To rate components include the logic to check for tax exemption eligibility.

Fastpath: For more information about how to make a customer tax exempt, refer to Contract - Tax Exemptions.

Some types of bill factors do not have the same value for all customers. Rather, their value may be unique for every customer or it may be unique for some customers and common for others. For any type of bill factor where a customer may have their own unique price, turn on **Value In Contract Term**. For any contract that has a unique value for a given bill factor, the bill factor and the contract's value must be indicated on the Contract page. When applying the rate for a bill factor of this type, the system will first check for the existence of a contract value for the contract and if one is not found, it will use the value on the bill factor. This field will be gray if a contract term referencing this bill factor exists. Variance Parameter bill factors may follow this same logic. Customers may override their Variance Parameter values by using a contract term.

Copyright © 2024, Oracle and/or its affiliates.

Note: The **Error if No Value** switch also affects what happens if there is no contract term for the bill factor. If the error switch is on, a billing error will be generated. If the error switch is off, no billing error will be generated and the rate component will be skipped.

Fastpath: For an example of a bill factor whose value is specified in a contract term, refer to Customer Specific Prices. For more information about how to specify a bill factor value in a customer's contract, refer to Contract - Rate Info.

Contract riders are used when the charge associated with the bill factor is only applicable to some customers for some period of time. For example, if a bill factor is used to provide a discount during the first 3 months of service, then a contract rider would be used. Turn on **Contract Rider Applicability** if the bill factor's applicability is contingent on the existence of a contract rider. This field will be gray if a contract rider referencing this bill factor exists on a contract.

The proration information tells the system how to handle changes to bill factor values and their optional contract riders that may occur during a customer's bill period. Turn on **Allow Proration** if changes to the bill factor's value and/or contract riders should be prorated. If the **Bill Factor Type** is not specified as `Interval` and if changes should **not** be prorated, indicate the **Rate Selection Date Algorithm** to use when the system retrieves bill factor values or determines if contract riders are applicable. Click here to see the algorithm types available for this system event.

Bill Factor Tree

The right half of this page is dedicated to a tree that shows the bill factors characteristics and their respective values (for `Regular` bill factors) or interval value Sets (for `Interval` bill factors). You can use this tree to both view high level information about these objects and to transfer to the respective page in which an object is maintained by using the available Go To buttons.

Where Used

Many types of Rate Components may use a Bill Factor to define the price being charged. Refer to Rate Component - Main Information for more information.

The `BILPC-TOTAL` algorithm type that is used to levy late payment charges references the bill factor that defines the late payment charge percent. Refer to *Contract Type - Main* for more information about this algorithm.

A contract references Bill Factors as follows:

- When a contract's `Apply To` rate component references tax exemptable bill factors AND the customer is tax exempt, the contract will define the bill factors from which the customer is exempt. Refer to Contract - Tax Exemptions for more information.
- When a contract's rate references bill factors whose prices are customer-specific (i.e., specified on a contract term), the contract will contain prices for these bill factors. Refer to Contract - Contract Values for more information.
- When a contract's rate references bill factors that are contingent on contract riders, the contract will define the rider period (if any) for these bill factors. Refer to *Contract - Contract Riders* for more information.

Register rules using the `Bill factor based UOM conversion` algorithm reference a bill factor to define the conversion amount.

Setting Up Bill Factor Characteristics

After defining general information about a bill factor, open **Main Menu, Rates, Bill Factor** and navigate to the **Bill Factor Characteristic** tab to define characteristic values for which a charge is applicable. For example, a bill factor used to levy the applicable city tax would require the definition of every city that levies a tax.

Note: If all values for a bill factor are specified in contract terms, you do not need to set up Bill Factor Characteristics.

Description of Page

Enter a **Characteristic Type** to indicate the type of characteristic that controls how the bill factor's value is defined. This field will be gray if Bill Factor Characteristics are specified in the following grid.

The **Characteristic Source** defines where the system retrieves the characteristic value at billing time.

The following table lists and describes the characteristics source for a bill factor:

Copyright © 2024, Oracle and/or its affiliates.

Characteristic Source	Description
`Account`	The characteristic type's value will be retrieved from the contract's account.
`Characteristic Collection`	Refer to Deriving / Passing In Characteristic Values for an explanation of this value.
`Main Person`	The characteristic type's value will be retrieved from the main person linked to the contract's account.
`N/A`	The characteristic type's value is the same for all customers using the rate. If this option is used, you must still choose a Characteristic Type. This Characteristic Type should have a single Characteristic Value. The ID of the characteristic value must be identical to the ID of the characteristic type.
`Contract`	The characteristic type's value will be retrieved from the contract.

Fastpath: For more information about characteristics, see *Setting Up Characteristic Types & Their Values* and An Illustration Of A Bill Factor And Its Characteristics.

Use the add and remove buttons to define the **Characteristic Values** that are relevant for the bill factor's Characteristic Type. For example, if the bill factor is used to levy a city tax, you would define the city's that have a city tax in the above collection.

Use the **Minutes Per Interval** if your Bill Factor is of type `Interval`. This defines, for each characteristic value, the number of minutes expected between each record of interval data.

Use the **External ID** if the values for this characteristic are interfaced to the system from an external source and you want to record an identifier for the source.

Note: If the bill factor's value is the same for all customers, you still must link a "dummy" characteristic value to the bill factor. In other words, you'll have to create a characteristic type called "n/a" and define for it a characteristic value of "n/a". Then, on the bill factor, you reference the "n/a" characteristic type and on the above window, you indicate a characteristic value of "n/a".

Defining Bill Factor Values

After the bill factor and its characteristic values are defined, you define the price / percent and GL distribution associated with each bill factor characteristic. The topics in this section describe the pages used to set up this information.

Bill Factor Value - Main

After the bill factor and its characteristic values are defined, select **Main Menu, Rates, Bill Factor Value** to define the price / percent applicable to each bill factor characteristic.

Note: Important! You do not set up Bill Factor Values for bill factors whose values are specified in an interval value data set or if all the values are customer specific an are defined as contract terms. Refer to Defining Interval Values for more information about interval value sets. Refer to Contract - Contract Value for more information about bill factors in contract terms.

Description of Page

Enter the **Effective Date** and respective **Value** for the bill factor and characteristic value. The first date should be the earliest date on which the bill factor's charges are effective. Subsequent effective dates and values are required whenever the value changes.

Note: A **Value** of 0 will be defaulted.

Copyright © 2024, Oracle and/or its affiliates.

Bill Factor Value - GL Distribution

Select **Main Menu, Rates, Bill Factor Value** and navigate to the **GL Distribution** tab to define the GL account affected by the bill factor characteristic.

> **Note: Important!** If you do not specify this information for a bill factor characteristic, the distribution code defined on the rate component that references the bill factor will be used.

Description of Page

If the bill factor characteristic's charges are booked to a different GL account depending on the characteristic value, specify the appropriate **Distribution Code** for each **Revenue Class**.

> **Note: Optional information.** Specifying distribution codes on a bill factor is optional. You would only do this if the GL distribution code differs based on the characteristic value (e.g., the city tax payable GL account differs depending on the city). If you do not enter the above information, the distribution code specified in the rate component that references the bill factor will be used.

Defining Interval Values

Open **Main Menu, Rates, BF Interval Values** to maintain interval values.

> **Note: Interval values must reference a bill factor.** You must set up a bill factor before you can define interval prices. Refer to Setting Up Bill Factors for more information.

Description of Page

Interval Value Set is a concatenation of important information about the data set. The **Interval Value Set ID** is a system-assigned random number that stays with an interval value set for the life of the system. This value is assigned when the set is first added.

All interval pricing information must reference a predefined **Bill Factor**, **Characteristic Type** and **Characteristic Value**. Use the **Bill Factor Characteristic** search button to select all three values. Refer to Setting Up Bill Factors for more information.

Use **Set Date / Time** to define the effective date of the set of prices or values defined in the following grid. This is considered a server date and will follow the seasonal time shifting as defined on the base time zone.

Use the **Start Date / Time** to indicate the starting point for the display of the interval data.

At least one interval data record must exist in the collection of interval data. For each interval data record, the **Interval Value Date / Time** and the **Interval Value** are captured. This date is considered a logical date and will follow the seasonal time shifting as defined on the bill factor / characteristic.

If seasonal time shift records are linked to both the base time zone and the bill factor/characteristic, the **Seasonal Time Shift** associated with your bill factor/characteristic is displayed. Refer to Setting Up Bill Factor Characteristics for more information.

If seasonal time shift records are linked to both the base time zone and the bill factor/characteristic, the **Seasonal Time Shift Remark** indicates whether the data is displayed and should be entered in legal time or standard time. If the data is displayed in legal time use the **Show in Standard** button to toggle to standard time. If the data is displayed in standard time use the **Show in Legal** button to toggle to legal time. If either the bill factor/characteristic or the base time zone do not indicate a seasonal time shift record, then the toggle button will not be visible and data is displayed and expected in standard time.

Where Used

Various rate component algorithms make use of interval pricing information.

Copyright © 2024, Oracle and/or its affiliates.

Bill Factor Interval Values Query

This query shows the most recent interval values for an interval value bill factor/characteristic and a time period as of a given time.

Open **Main Menu, Rates, BF Interval Values Query** to display interval bill factor values.

Description of Page

Information describing the **Bill Factor, Characteristic Type** and **Characteristic Value** whose data you wish to query is displayed.

Use the **Start Date / Time** and **End Date / Time** if you want to limit the query to a specific time period.

Use the **Set Date / Time** to indicate the "as of what date" to use for retrieving interval data. Only data sets with a set date/time prior to this date/time will be accessed.

The **Selection Type** is used to determine what data to show

`History` This option will display all bill factor interval data for the Bill Factor/Characteristic and time period, if specified. It will include data in data sets of any status value.

`Most Recent Complete Data` This will find the most recent bill factor interval value data for each interval from completed data sets only.

Once you have selected your criteria, use the search button to display the data. The interval value data collection will contain the following information:

The **Interval Value Date/Time** and **Interval Value** for each interval is displayed. Adjacent, information about the data set to which this interval is linked is displayed, including the **Set Date / Time** and **Interval Value Set ID**.

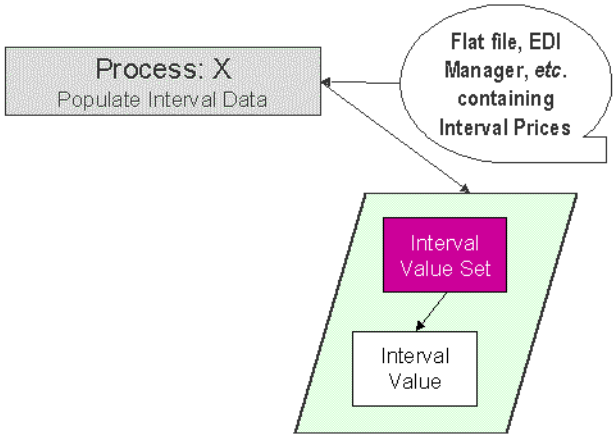
If seasonal time shift records are linked to both the base time zone and the bill factor/characteristic, the **Seasonal Time Shift** associated with your bill factor/characteristic is displayed.

The **Seasonal Time Shift Remark** indicates whether the data is displayed and should be entered in legal time or standard time. If the data is displayed in legal time use the **Show in Standard** button to toggle to standard time. If the data is displayed in standard time use the **Show in Legal** button to toggle to legal time.

Note: If seasonal time shift information is missing for the bill factor/characteristic or for the base time zone, then data is always displayed and expected in standard time.

Upload Interval Values

The following diagram illustrates the processes involved in the uploading of interval values into the system.



The topics in this section describe how these processes work.

Copyright © 2024, Oracle and/or its affiliates.

Process X - Populate Interval Data

Process X refers to the mechanism used by your organization to populate the Interval Value tables. Process X will have the following responsibilities:

- It will determine the appropriate Bill Factor and Characteristic to which this data belongs. Recall that the bill factor / characteristic may reference an external id.
- It will create a new Interval Value Set for this bill factor characteristic with the current date and time
- If the interval data is from a different time zone, or is not in standard time for the base time zone, the time of each interval must be converted to the base time.
- All prices being uploaded for the same bill factor and characteristic should be inserted into the Interval Value table and should reference the same Interval Value Set
- Inserts a record in the interval value set key table.

The topics in this section describe the tables to populate.

Interval Value Set

You must create a data set for each batch of records being uploaded in the system for the same bill factor / characteristic. The name of this table is CI_INTV_VAL_SET .

Table 1: - CI_INTV_VAL_SET Table Columns

The following table lists and describes the columns of the **CI_INTV_VAL_SET** table:

Column Name	Length	Req'd	Data Type	Comments
INTV_VAL_SET_ID	12	Y	A/N	This is a random unique key. This key is referenced on the interval value record. It's suggested that this is a random number. If your process that inserts records on this table is capable of calling a COBOL routine, call CIPZKEYG and it will supply a 12 digit random number for you.
BF_CD	8	Y	A/N	This along with CHAR_TYPE_CD and CHAR_VAL indicate the bill factor / characteristic to which this interval value set belongs. Refer to Setting Up Bill Factors for more information.
CHAR_TYPE_CD	8	Y	A/N	This along with BF_CD and CHAR_VAL indicate the bill factor / characteristic to which this interval value set belongs. Refer to Setting Up Bill Factors for more information.
CHAR_VAL	16	Y	A/N	This along with BF_CD and CHAR_TYPE_CD indicate the bill factor / characteristic to which this interval value set belongs. Refer to Setting Up Bill Factors for more information.

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Length	Req'd	Data Type	Comments
SET_DTTM	26	Y	DateTime	This is the data/time that the data set was received. This must be populated with the current date / time. This is considered a server date and will follow the seasonal time shifting as defined on the base time zone.

Interval Value

Each piece of interval data for the same bill factor/characteristic will be stored as an Interval Value record linked to the newly created data set. The name of this table is CI_INTV_VAL .

Table 2: - CI_INTV_VAL Table Columns

The following table lists and describes the columns of the **CI_INTV_VAL** table:

Column Name	Length	Req'd	Data Type	Comments
INTV_VAL_SET_ID	12	Y	A/N	This is a foreign key reference to the Interval Value Set that groups the interval values together. This should be your newly created id from
INTV_VAL_DTTM	26	Y	DateTime	This is the date and time of the interval value. This must be stored relative to the base time zone.
INTV_VAL	18.7	Y	N	This is the value (price) for the interval.

Interval Value Set Key

When a system-generated key value is assigned to a record, you need to also store the key value in a key table that corresponds to the record's database table. The name of this table is CI_INTV_VAL_SET_K .

Table 3: - CI_INTV_VAL_SET_K Table Columns

The following table lists and describes the columns of the **CI_INTV_VAL_SET_K** table:

Column Name	Length	Req'd	Data Type	Comments
INTV_VAL_SET_ID	12	Y	A/N	This should be your newly created Interval Value Set ID from above.
ENV_ID	6	Y	N	This is the environment id defined on the installation options.

Defining Bill Factor Variance Parameter Values

Open **Main Menu, Rates, BF Variance Parameter Values** to maintain variance parameter bill factor values.

> **Note: Variance Parameter Bill Factor values must reference a bill factor and characteristic.** You must set up a bill factor and its characteristics before you can define variance parameter values. Refer to Setting Up Bill Factors for more information.

Description of Page

Copyright © 2024, Oracle and/or its affiliates.

The **BF Characteristic Variance Parameter Group** search button gives you the ability to search for existing Variance Parameter BF values for a Bill Factor, Characteristic and Variance Parameter Group combination.

The **BF Characteristic** search button gives you the ability to search for a Bill Factor, Characteristic combination to which the Variance Parameter values will be linked.

Once a bill factor and characteristic combination are chosen, the system displays the values of the **Bill Factor**, **Characteristic Type**, **Characteristic Value** and **Value Type**.

Choose the **Variance Parameter Group** to determine the collection of Variance Parameter to which prices will be defined.

Indicate the **Effective Date** of the collection of **Variance Parameter** and their **Values**.

Where Used

Various rate component algorithms make use of Variance Parameter value information. This information is also available for any type of algorithm that may need to use the information, for example interval data derivation algorithms.

Defining Rate Versions

After defining general information about a rate on Rate Schedule Maintenance, you must link to it a rate version. The rate version defines the effective date of the calculation rules defined in its rate components. A rate schedule will have multiple rate versions if its calculation rules (i.e., its rate components) change over time.

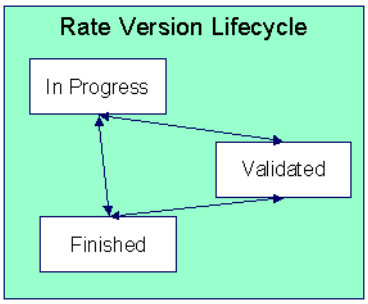
Note: A switch on the rate schedules controls what happens if multiple rate versions are effective during a bill period. When the system creates a bill segment for a contract, it checks if multiple rate versions are in effect during the bill period. If so, it uses the rate schedule's **Allow RVs Proration** switch and **RV Selection Date** flag to determine if it should prorate the various rate versions or if it should pick one of the rate versions effective during the bill period.

After a rate version exists, you add rate components to it using Rate Component Maintenance and/or Rate Version Merge. After you have added all necessary rate components, don't forget to return to Rate Version - Main and change the state of the rate version to `Finished` (otherwise, it cannot be used by billing).

The topics in this section describe how to setup a rate version.

Lifecycle of a Rate Version

The following diagram shows the possible lifecycle of a rate version (you'll notice that you can change a rate version from any state to any state).



A rate version is in the `In Progress` state while you build its rate components. While a rate version is in this state, the system does not perform cross validation between rate components as they are added / changed / deleted using Rate Component Maintenance and/or Rate Version Merge. However, the system will perform inter-field validation applicable to each type of rate component. Rate versions in this state may not be used by Rate Check, nor will they be used up by the billing process.

Copyright © 2024, Oracle and/or its affiliates.

You transition a rate version to the `Validated` state when you're ready to check it using Rate Check, but you don't want billing to use it to calculate charges. When a rate version is transitioned to this state, the system performs cross validation between its rate components. If any validation errors are found, an error message is displayed and the status remains `In Progress`. If all validation checks are passed, the rate version status will become `Validated`. We'd like to stress that rate versions in this state may be used by Rate Check, but will *not* be used by billing.

You transition a rate version to the `Finished` state when it can be used by billing to calculate charges. All of the cross validation between rate components will be performed when transitioning to this state.

Note: Finished and Validated states also affect rate component maintenance. If a rate version is `Finished` or `Validated`, the system performs internal consistency checks every time a rate component is added, changed or deleted using Rate Component Maintenance and/or Rate Version Merge. If you don't want these checks to be performed while you're working on a rate version's rate components, change its state to `In Progress`.

Rate Version - Main

Use **Main Menu, Rates, Rate Version** to maintain a rate version.

Description of Page

Rate Version contains basic information about the rate version. This information only appears after the rate version has been added to the database.

Rate Schedule defines the rate to which the rate version is linked. **Effective Date** defines the date on which the rate version's rate components become effective. These fields are the unique identifier of the rate version. Both fields become protected after the rate version is added to the database.

Note: Multiple rate versions may be prorated. When the system creates a bill segment for a contract, it checks if multiple rate versions are in effect during the bill period. If so, it uses the rate schedule's **Allow RVs Proration** switch and **RV Selection Date** flag to determine if it should prorate the various rate versions or if it should pick one of the rate versions effective during the bill period.

The **Rate Version Status** indicates the current state of the rate version. Refer to Lifecycle of a Rate Version for more information.

CAUTION: Billing ignores rate versions that are not in the `Finished` status. This means that if you create a new rate version and forget to finish it, billing will use the prior rate version when it calculates the customers' bills.

Specify the description of the rate version in **Description on Bill**. This description is saved on bill calculation details created using this rate version.

The following table lists and describes the substitution variable that you can use in the **Description on Bill** field:

Substitution Character	What It Does
%D	Replaces the substitution character with message 10101 from message category 4. This message is maintained in the standard message table. We recommend this message be set to "(prorated for %1 of %2 days)" where %1 will be substituted with the number of days of prorated product/charge type charged on the bill line and %2 will be the "normal" days associated with the rate's prices. This message will only be substituted when the rate version is prorated.

The following table illustrates how you can define a description on a bill:

Description on Bill	How It Looks On A Bill
ACH service %D	ACH service (prorated for 10 of 30 days)

Copyright © 2024, Oracle and/or its affiliates.

The tree that follows contains a summary of the rate version's rate components. You can use this tree to both view high-level information about these objects and to transfer to the respective page on which an object is maintained.

> **Note:** The duplicate action in the action button bar enables you to duplicate a rate version and its rate components. Refer to Duplicate Button in the system wide standards document for more information.

Rate Version - Bill Print Info

This page allows you to maintain the bill print information for all of the rate version's rate components. Note, you can also maintain this information for individual rate components on Rate Component - Main.

Description of Page

The grid contains an entry for each rate component linked to the rate version. The following information is displayed in the grid:

Sequence uniquely identifies the rate component. It also controls the order in which the rate component is processed when the rate version is processed by billing.

Description is the rate component's description.

Description on Bill is the verbiage that appears on the bill line generated for this rate component. You can dynamically substitute variables into this description (e.g., price and quantity) when billing calculates the rate component's charges. Refer to How To Use Description on Bill for a discussion of the various substitution variables.

Turn on the **Print** switch if you want the bill line to appear on the customer's bill.

> **Note: Some bill lines don't print.** You may wonder why we give you the ability to not print a bill line. The reason is because ALL bill lines are shown on Bill Segment - Calculation Details. However, only those lines marked as Print are presented to the customer. This way, if you want to suppress some calculation details, you can, but still show the details to the customer service rep.

The **Print If Zero** switch is only enabled if the **Print** switch is turned on. It allows you to indicate if a bill line should print when the bill line's value is zero:

- Turn this switch on if the bill line should print when the calculated value is zero.
- Turn this switch off if the bill line should not print when the calculated value is zero.

> **Note:** The rate component is created in the bill segment calculation line, but an indicator exists to tell the extract program that this should not be included.

How To Use Description on Bill

Enter the verbiage to appear on the bill segment in the **Description On Bill**. Turn on **Print** if the verbiage should appear on the printed bill.

> **Note: Suppressing calculation details on the printed bill.** Only rate components used to calculate intermediate values are typically suppressed on a bill. For these types of rate components, we recommend using **Description on Bill** to show users how the line was calculated, but turn off the **Print** switch. This way, users will see the calculation details on the Bill Segment - Calc Lines window, but the customer won't see it on the bill.

Many lines that appear on a bill do not contain static verbiage. Rather, the printed lines contain information from the calculation process. For example, the bill line for state tax would typically contain the tax percent.

You control exactly what is substituted in the bill line by entering one or more substitution variables in the **Description on Bill** field.

The following table lists and describes the various substitution variables that you can use in the **Description on Bill** field:

Copyright © 2024, Oracle and/or its affiliates.

Substitution Character	What It Does
%A	Replaces the substitution character with the amount of the bill line. Please be aware of the following: • If the rate component isn't designated as being "for calculation purposes only" (FCPO), all trailing zeroes up to the decimal position of the rate's currency will be removed. If the rate component is designated as being "for calculation purposes only" (FCPO), the number of decimal places as defined in the rate component's precision will be shown. • All leading zeroes will be removed unless the value is between 1 and -1 in which case a leading 0 will appear before the decimal place. • The Symbol on the Currency control table will prefix the value. • The position of the negative sign and currency symbol is dependent on the display profile of the user who generate the bill (or who submitted the batch billing process).
%B	Replaces the substitution character with the begin/start date covered by the bill calculation line. If the charge is proratable, %B may be used in conjunction with %E to display the period covered by the bill calculation line.
%C	Replaces the substitution character with the number of items being billed. Leading zeroes are suppressed.
%D	Replaces the substitution character with message 10101 from message category 4. This message is maintained in the standard message table. We recommend this message be set to "(prorated for %1 of %2 days)" where %1 will be substituted with the number of days of prorated product/charge type charged on the bill line and %2 will be the "normal" days associated with the rate's prices. This message will only be substituted when the rate component is prorated. Note, neither %1 nor %2 are required. Either this character OR %F should be used when you desire to substitute the price / percent into the bill line. If the charge is proratable, %D should be used in conjunction with %R to print an appropriate message when the charges are prorated.
%E	Replaces the substitution character with the end date covered by the bill calculation line. If the charge is proratable, %E may be used in conjunction with %B to display the period covered by the bill calculation line.
%F	Replaces the substitution character with message 10102 from message category 4. This message is maintained in the standard message table. We recommend this message be set to "(prorated by %1)" where %1 will be substituted with the factor used to prorate the charge. This message will only be substituted when the rate component is prorated. Note, %1 is not required. If the charge is proratable, %F should be used in conjunction with %I to print an appropriate message when the charges are prorated.
%I	Replaces the substitution character with the value specified in the rate component / bill factor. If the value is a percent, • all leading and trailing zeroes will be suppressed, • and a "%" will be suffixed to the resultant value.

Copyright © 2024, Oracle and/or its affiliates.

Substitution Character	What It Does
	If the value is not a percent, • all trailing zeroes up to the decimal position of the rate's currency will be removed, • and all leading zeroes will be removed unless the value is between 1 and -1 in which case a leading 0 will appear before the decimal place, • and note that the position of the negative sign and currency symbol is dependent on the display profile of the user who generate the bill (or who submitted the batch billing process). Either this character OR %R should be used when you desire to substitute the price / percent into the bill line. Using %I will cause the value as specified in the rate component to be substituted regardless of whether the charges are prorated. If the charge is proratable, %I should be used in conjunction with %F to print an appropriate message when the charges are prorated.
%O	Replaces the substitution character with the description of the Variance Parameter being billed on this line
%Q	Replaces the substitution character with the service quantity being billed. Leading zeroes are suppressed. The decimal places shown are specified in a field on the Unit Of Measure control table.
%R	Replaces the substitution character with the value specified in the rate component / bill factor. Either this character OR %I should be used when you desire to substitute the price / percent into the bill line. Using %R causes the prorated price / percent to be printed in the bill line (%I causes the price / percent as specified in the rate to be printed during proration situations). If the charge is proratable, %R should be used in conjunction with %D to print an appropriate message when the charges are prorated.
%S	Replaces the substitution character with the summed amount. All trailing zeroes up to the decimal position of the rate's currency will be removed. All leading zeroes will be removed unless the value is between 1 and -1 in which case a leading 0 will appear before the decimal place. The Symbol on the Currency control table will prefix the value. If the value is negative, a "-" will prefix the value. This character should only be used with Minimum, Apply To, and Summary rate components.
%T	Replaces the substitution character with message 10103 from message category 4. This message is maintained in the standard message table. We recommend this message be set to "(%1% exempt)" where %1 will be substituted with the tax exemption percent as specified in the contract. This message will only be substituted when the rate component's bill factor is eligible for tax exemptions AND the customer is exempt from the tax. Note, %1 is not required.
Lines that don't print	If the bill line is not printed (as defined by the Print switch), message 10104 from message category 4 will be appended to the bill line. This message is maintained in the standard message table. We recommend this message be "NOT SEEN BY CUSTOMER!" This message appears in the bill line that appears in the billing windows but does not appear on the customer's bill.

The prorated references will only appear if the rate component's charges are prorated due to a short or long bill period.

Copyright © 2024, Oracle and/or its affiliates.

Designing Rate Components

Rate components are the fundamental building blocks of your bill calculation algorithms. All of the other rate and bill factor set up tasks are done in preparation of the creation of these records. This means that you must design your rate components before you can design bill factors, register rules, SQ rules, SQ identifiers, eligibility rules. We strongly recommend that you design "on paper" how every rate component looks on every rate schedule before you attempt to set up any of the rate control tables.

Fastpath: All of the calculation details and financial information that appear on a bill segment are derived using information on rate components. If you want to see the results of what rate components do, refer to Bill Segment - Calc Lines.

CAUTION: There are innumerable ways to design rate components for a given rate. Some designs will result in easy long-term maintenance; others will result in maintenance headaches. In this section, we provide information to help you understand the ramifications of the various options. Before you set up your production rate components, we encourage you to gain intuitive understanding of these options by using the system to prototype the alternatives.

The topics in this section provide background information that will facilitate the construction of your rate components.

Rate Component Design Methodology

Designing rate components is an iterative process. Over time, you will develop intuitive skills that will allow you to avoid some iterations. However, when you're starting out, we recommend you follow these steps to design your rate components:

- Obtain copies of existing bills that use the rate in question. If the rate is new, then write up EXACTLY how the information should appear on the customers' printed bills. If the rate has optional charges (e.g., charges that only apply after the customer has consumed a large volume), make sure your examples encompass every possible scenario.
- Next, look at the bill lines and ask yourself what are the variables that cause each line's charges to be calculated the way they are. Consider the following examples:
  - "Monthly service charge: $50.00". This is a simple service charge that appears regardless of how much the customer consumed. This line doesn't really have any variables that affect its amount. Rather, the amount is a constant value of $50.00.
  - "Minimum monthly charge of $100.00". This is a charge that only appears if the total of the prior lines is less than $100.00. The variables involved in this line are a little complicated because they must be calculated at billing time by adding up several other lines and comparing them to the minimum charge amount. This means the variables are the total of prior lines and the minimum charge amount.
  - "1.25% County Surcharge". This charge is similar to the minimum charge in that it is calculated by adding up one or more prior lines and applying a percentage to the sum. This means the variables are the total of prior lines plus the tax percentage.
- After you've determined all of the potential bill lines you can start designing your rate components. Typically you create one rate component for every line that can appear on the customer's bill. When you create a rate component, you will categorize it as one of the following types:

The following table lists and describes the types of rate components:

Type of Rate Component	Description
`Flat Charge`	This type of rate component is used to create bill lines that levy service charges and fees that aren't based on how much the customer uses. The monthly service charge shown above would be levied using this type of rate component.
`Service Quantity`	This type of rate component is used to create bill lines that levy charges based on usage.

Copyright © 2024, Oracle and/or its affiliates.

Type of Rate Component	Description
	You'll find it helpful to differentiate your service quantity rate components into two categories - stepped and non-stepped. Stepped charges are those where there's a different price for different ranges of consumption (e.g., the first 50 consumed quantities are charged at $0.43 per quantity, anything in excess of 50 consumed quantity is charged at $0.71 per quantity). We recommend doing this because a) you frequently show a subtotal of the steps on the customer's bill, b) steps, by definition, mean you have several rate components that are dependent on each other, and c) they frequently involve complex calculations in order to define the step boundaries.
`Apply To`	This type of rate component is used to create bill lines that levy charges based on the amount calculated on other bill lines. The county surcharge shown above would be levied using this type of rate component.
`Summary`	This type of rate component is used to create a "subtotal" on the bill. It exists purely for aesthetic purposes.
`Minimum Charge`	This type of rate component is used to create bill lines that levy charges when the sum of previously calculated lines is less than the minimum charge amount. The minimum monthly charge shown above would be levied using this type of rate component. Note that if the values being compared are negative values, the comparison is NOT done on the absolute values, but rather on the actual values. For example, imagine you have a minimum discount of $ -2.00 and previous rate components have calculated a discount of $ -1.00. You want a rate component to create a bill line for $ -1.00 to apply a further discount. A minimum charge rate component will not work in this case because -1 is considered more than -2. For this business scenario you should use a maximum charge rate component.
`Maximum Charge`	This type of rate component is used to create bill lines that levy charges when the sum of previously calculated lines is more than the maximum charge amount. Note that if the values being compared are negative values, the comparison is NOT done on the absolute values, but rather on the actual values. For example, imagine you have a maximum discount of $ -2.00 and previous rate components have calculated a discount of $ -3.00. You want a rate component to create a bill line for $1.00 to reduce the discount. A maximum charge rate component will not work in this case because -3 is considered less than -2. For this business scenario you should use a minimum charge rate component. Note - these types of rate components are rare. You tend to only see them in deregulated markets where customers have agreed to pay the market rate but want a cap on the maximum they'll pay each bill period.
`Exact Charge`	This type of rate component is used to force a bill to add up to a given amount (regardless of how much the bill would be based on earlier rate components). Note - these types of rate components are rare. You tend to only see them in deregulated markets during the transition period between regulated rates and complete deregulation. In this situation, you show on the bill what the charges would be, given a deregulated environment, but then adjust the bill back to an amount equal to the regulated rate. This charge is called a Competition Transition Charge (CTC) in California.

Copyright © 2024, Oracle and/or its affiliates.

Type of Rate Component	Description
`Calculation Algorithm`	This type of rate component enables you to produce bill calculation lines based on logic in an algorithm that you supply. Use this rate component when none of the other rate component functions will provide you with the logic you require.
`Interval Pricing`	This type of rate component is used to apply interval prices to interval quantities. See *Interval Pricing Rate Application* for more information.
`Variance Parameter Pricing`	This type of rate component is used to map and/or price interval quantities into Variance Parameter quantities given a Variance Parameter map. See *Variance Parameter Mapping and Pricing* for more information.

- For each rate component, determine if the customer must meet some form of eligibility criteria before the rate component is applied. For example, you may have a rate component that calculates an employee discount. This rate component may have eligibility criteria that requires the person to have a given characteristic value before the rate component is applied. Refer to The Big Picture of Rate Component Eligibility for more information.
- For each type of rate component (except summary), determine how you'll define its unit rate / percent / flat amount. You have four choices:
  - Specify the value directly in the rate component.
  - Tell the rate component to use the value defined in a *bill factor*.
  - Tell the rate component to use the value calculated by a " *for calculation purposes only*" rate component.
  - Tell the rate component to call an algorithm. The algorithm will return the value.

The following table lists and describes the sources from where you can derive the rate component value:

Source of Rate Component Value	Description
`Specify value in the rate component`	You'd specify the value in the rate component when the value in the rate is specific to that rate and is applied to all customers regardless.
`Use a bill factor to define the value`	The following paragraphs describe when you'd use a bill factor to define a unit rate / percent / flat amount. The same charge exists in many rates. For example, if there are 3 rates that all contain the same service charge, it would make sense to use a bill factor to levy this charge rather than specify the same value on 3 rate components. The amount being charged is dictated by some external organization and therefore can change independently from the rate. For example, tax percentages and use fees are set by taxing authorities and can be changed at any time (and you don't want to change the rate when these charges change). The amount being charged varies depending on WHERE the customer lives. If you have a rate where the price per gallon differs depending on where the customer lives, you should use a bill factor to levy this type of charge. A charge (or discount) is only levied for a subset of customers for defined periods of time. For example, if you have a discount that is only given to customers who have had service for more than 2 years, you would use a bill factor to specify the discount percent. The price differs per customer. For example, you may have a rate for customers where the price of a product is different for

Copyright © 2024, Oracle and/or its affiliates.

Source of Rate Component Value	Description
	every customer. In this situation, you would use a bill factor to specify the price.
`Use a "for calculation purposes only" rate component`	This is complicated to explain and is not used very often, but it is very powerful and handles some very complicated algorithms. It works like this - you calculate the price / percent / flat charge on another rate component (marked as "for calculation purposes only" so it won't affect the bill amount), and then reference it on the "real" rate component. See the Calculated Minimum Charges example for an illustration of when this method is useful.
`Call an algorithm`	This is complicated to explain and is not used very often, but it is very powerful and handles some very complicated examples. It works like this - you call an algorithm and it calculates the price / percent / charge. One example that would require the use of an algorithm would be when the price differs based on the total amount consumed. For example, if the customer uses less than 10 units, the price is $1.50; whereas if the customer uses more than 10 units, the price is $2.00. We supply an example of one such algorithm in the base package. If your rates require additional algorithms, you will have to develop additional algorithms. Refer to `How to set up service quantity rate components` under Rate Component - Main Information for more information about this type of algorithm.

- After you've categorized the bill lines into the various types and know how the price is defined, you determine how each line's charges are reflected in your general ledger. You will need to work with your accounting department as they will tell you the exact revenue, expense, and liability accounts that will be affected by the charges.
- Finally, read the provisions of your rate (the legal wording) and make sure you haven't left out something. This may involve having to discuss confusing provisions with your legal department.

At this point, you're ready to start entering your rate components.

Note: Examples. Refer to Rate Examples for many sample rate components.

Rate Component Rounding

The topics in this section describe how to control how each rate component's final value is rounded.

Rounding Precision Is Defined On A Rate Component

A rate schedule references a currency code. A rate's currency code defines its maximum number of decimal places. These decimal places, in turn, control the smallest unit to which *most* rate components can be rounded (i.e., the rate component's precision). For example,

- If a rate's currency has 2 decimal places, the smallest rounding precision that *most* of its rate components can have is 0.01. In other words, the smallest unit to which *most* dollar-based rate components can be rounded is to the cent. Note, whether you round up / down / to the nearest is discussed below.
- If a rate's currency has no decimal places, the smallest rounding precision that *most* of its rate components can have is 1.00.

Copyright © 2024, Oracle and/or its affiliates.

The system supports the notion of a rate component having a greater rounding precision than its currency. For example, a rate's currency may support 2 decimal places, but you can setup rate components to have a rounding precision of 0.05.

The reason that we underlined the word *most* in the previous paragraphs is that these rules do not apply to "for calculation purposes only" (FCPO) rate components. FCPO rate components don't contribute real amounts to a bill (they exist to calculate intermediate results that are used by later rate components) and therefore they can support a rounding precision of up to 0.00001 (5 decimal places).

Rounding Method Is Defined On A Rate Component

When you setup a rate component you must define the rounding method. You are given the following choices:

- `Always round up`. This rounding method rounds a rate component *up* to the nearest value (as controlled by the rate component's precision). For example, if a rate component's precision is 0.01 and a value of 0.011 is calculated, the rate component's final value would be rounded up to 0.02.
- `Always round down`. This rounding method rounds a rate component *down* to the nearest value (as controlled by the rate component's precision). For example, if a rate component's precision is 0.01 and a value of 0.019 is calculated, the rate component's final value would be rounded down to 0.01.
- `Round to the nearest value`. This rounding method rounds a rate component to the nearest value (as controlled by the rate component's precision). For example, if a rate component's precision is 0.01 and a value of 0.019 is calculated, the rate component's final value would be rounded up to 0.02. Whereas, if a value of 0.012 is calculated, the rate component's final value would be rounded down to 0.01.

Rounding And FCPO Rate Components

As described above, for calculation purpose only (FCPO) rate components don't contribute real amounts to a bill. FCPO rate components exist to calculate intermediate results that are used by later rate components. We therefore allow FCPO rate components to have a rounding precision greater than the rate's currency code to a maximum of 0.00001 (5 decimal places).

Please be aware of the following:

- As described under How To Use Description on Bill, you can indicate that the final amount of a bill line should be substituted into the bill line's description by using the %A substitution variable in the rate component's **Description On Bill**. If you do this for an FCPO rate component, the system will show the number of decimal places as dictated by the FCPO's precision.
- While rate application calculates bill lines, the system maintains each rate component's value in memory. For FCPO rate components, this means that precisions greater than.01 are available during rate application. For example, you could indicate an FCPO rate component has a precision of.0001 and the resultant value will be maintained in memory and is available to other rate components while rate application executes.
- When rate application completes, the FCPO's value will be either saved in the bill line's amount or discarded as per the value of the rate component's **FCPO retention rule**. If you indicate that the FCPO amount should be retained on the bill line and its precision is greater than 2 decimal places, the FCPO amount will be rounded to two decimal places before it's saved on the database. Keep in mind that this could result in an inconsistency if you used the %A substitution variable in the rate component's **Description On Bill** as the value substituted into the bill line will have a greater precision.

Interim Rounding For Apply To Rate Components

The calculations for Apply To rate components involve several steps with interim values stored along the way. For every interim value that is stored, rounding rules are applied. The following steps describe the interim rounding rules:

- If the value type is not `Unit Rate`, the amounts for all the "cross-reference" rate components are added together and stored in the database as the Base Amount. The base amount is stored with a precision of.01 and it is rounded to the nearest value based on the currency's precision.
- The rate component calculates its charge and keeps the result in a temporary field with a precision of.00001.
- The remaining steps depend on whether or not a tax exemption is applicable

Copyright © 2024, Oracle and/or its affiliates.

- If NO tax exemptions are applicable, the temporary field is rounded based on the rate component's rounding rules to produce the final amount.
- If there are tax exemptions, the temporary field is rounded to the nearest.01 decimal precision prior to applying the tax exemptions. When the tax exemption is applied, the result is rounded according to the currency precision and stored as the Exempt Amount. The system calculates the final amount by subtracting the exempt amount from the rounded temporary amount and then applying the rate component's rounding rules.

Rounding At The End

The following is an example of a simple bill segment with three bill lines.

Bill Segment - Contract 1

Line Description	Amount	GL Impact
Usage charge	$501.00	Revenue $501.00
Tax	$5.01	Tax payable $5.01
Total	$506.05	Rounding $0.04

The first two bill lines are calculated using simple rate components that have a precision of.01. The last bill line is special, it rounds the bill up to the next highest 0.05 (this is an example from a country that doesn't have cents, their smallest coin is 0.05). The following would be necessary to calculate the last bill line:

- The rate component would be an exact charge rate component
- Its value (i.e., the exact charge) would be calculated using a value calculation algorithm that sums the first two rate components and rounds them up to the nearest.05. Refer to RCVL-RNDXRF for a value calculation algorithm that can perform this function.

An alternative approach is to do the following:

- Use a summary rate component to calculate the exact charge. This rate component would require the following characteristics:
  - FCPO
  - Non printing
  - Rounding type would be round up
  - Precision would be 0.05
- Reference the summary rate component as the exact charge rate component's value

The Big Picture Of Rate Component Eligibility Rules

You can use eligibility rules to define the criteria when a rate component should be applied (or skipped). For example,

- You may have a rate component that calculates an employee discount. This rate component may have eligibility criteria that require the person to have a given characteristic type and value (this assumes you use a person characteristic to identify employees) before the rate component is applied.
- You may have a rate component that calculates a surcharge if the customer uses 25% more than they used in the same period in prior year.

Note: Eligibility rules are optional. If you put eligibility rules on a rate component, the rate component will be skipped or applied as per the instructions in the rules. If you don't put eligibility rules on a rate component, the rate component will be executed (i.e., rate components are eligible by default).

The topics in this section describe eligibility rules.

Copyright © 2024, Oracle and/or its affiliates.

A Rate Component Is Eligible By Default

If you don't want to setup eligibility rules, you don't have to. If you don't specify eligibility rules on a rate component, it's eligible (i.e., it's going to be processed by rate application).

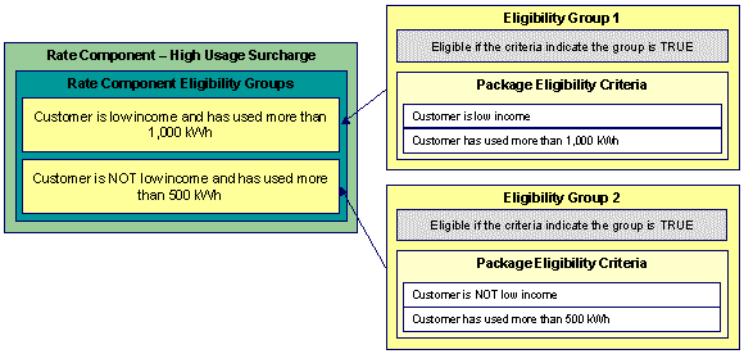
Criteria Groups versus Eligibility Criteria

Before we provide examples of eligibility criteria, we need to explain two concepts: Criteria Groups and Eligibility Criteria. A rate component's criteria groups control whether rate application executes a rate component. At a high level, it works like this:

- A criteria group has one or more eligibility criteria. A group's criteria control whether the group is considered TRUE or FALSE.
- When you create a group, you define what should happen if the group is TRUE or FALSE. You have the following choices:
  - The rate component should be applied
  - The rate component should be skipped
  - The next group should be checked

We'll use the following example to help illustrate these points. Assume a rate component is only eligible if:

- The customer is a low income customer and has made more than 1000 transactions.
- OR, the customer isn't a low income customer and has made more than 500 transactions.



This rate component requires two eligibility groups because it has two distinct conditions:

- IF (Customer is low income AND the customer has used more than 1,000 transactions)
- IF (Customer is NOT low income AND the customer has used more than 500 transactions)

If either condition is true, then the rate component is eligible.

The following table illustrates the criteria groups that you need to set up to support the above requirements:

Group No.	Group Description	If Group is TRUE	If Group is FALSE
1	Customer is a low income customer and has used more than 1,000 transactions	`Eligible`	`Check next group`
2	Customer is NOT a low income customer and has used more than 500 transactions	`Eligible`	`Ineligible`

The following table describes the eligibility criteria defined for the first criteria group:

Copyright © 2024, Oracle and/or its affiliates.

Group 1: Customer is a low income customer and has used more than 1,000 transactions				
Seq	**Logical Criteria**	**If Eligibility Criteria is TRUE**	**If Eligibility Criteria is FALSE**	**If Insufficient Data**
10	Customer is low income	`Check next condition`	`Group is false`	`Group is false`
20	Customer has used more than 1,000 transactions	`Group is true`	`Group is false`	`Group is false`

The following table describes the eligibility criteria defined for the second criteria group:

Group 2: Customer is NOT a low income customer and has used more than 500 transactions				
Seq	**Logical Criteria**	**If Eligibility Criteria is TRUE**	**If Eligibility Criteria is FALSE**	**If Insufficient Data**
10	Customer is NOT low income	`Check next condition`	`Group is false`	`Group is false`
20	Customer has used more than 500 transactions	`Group is true`	`Group is false`	`Group is false`

The next section describes how you'd setup the specific logical criteria in each of the groups.

Defining Logical Criteria

When you setup an eligibility criterion, you must define two things:

- The field to be compared
- The comparison method

You have the following choices in respect of identifying the ***field to be compared*** :

- You can retrieve a characteristic value linked to any of the following:
  - The contract being billed
  - The contract's account
  - The main person linked to the contract's account
  - In addition, you can also use a characteristic value that is derived while the rate is being calculated (characteristic values can be created by SQ rules, register rules and many other rate component algorithms)
- You can retrieve the value of a given service quantity
- You can retrieve the final value of an earlier rate component
- You can execute an algorithm to retrieve a field value from someplace else in the system. This is a very powerful feature, but it's not terribly intuitive. We'll present a few examples later in this section to illustrate the power of this approach.

You have the following choices in respect of identifying the ***comparison method*** :

- You can choose an operator (e.g., >, <, =, BETWEEN, IN, etc.) and a comparison value.
- You can execute an algorithm that performs the comparison (and returns TRUE, FALSE or INSUFFICIENT DATA). This is also a very powerful feature, but it's not terribly intuitive. We'll present a few examples later in this section to illustrate the power of this approach.

The Examples Of Rate Component Eligibility Rules provide examples to help you understand this design.

Examples Of Rate Component Eligibility Rules

The topics in this section provide examples about how to setup rate component eligibility rules.

Copyright © 2024, Oracle and/or its affiliates.

A Rate Component With A Service Quantity Comparison

Imagine a rate component that is only eligible if the customer is a low-income customer and has used more than 1,000 consumed quantity of a product. This rate component would need the following eligibility rules:

- Customer is a low income customer
- The customer has used more than 1,000 quantity of a product during the bill period

These rules require only one eligibility group on the rate component.

The following table illustrates the eligibility group defined on the rate component:

Group No.	Group Description	If Group is TRUE	If Group is FALSE
1	Customer is a low income customer and has used more than 1,000 quantity of a product	`Apply rate component`	`Skip rate component`

The following table illustrates the criteria defined in the eligibility group:

Group 1: Has banking service and user is part of user group A, B or C					
Seq	**Field to Compare**	**Comparison Method**	**If TRUE**	**If FALSE**	**If Insufficient Data**
10	Main person characteristic: Characteristic type = Low income indicator	= YES	`Check next condition`	`Group is false`	`Group is false`
20	SQ value: UOM = product	>= 1,000	`Group is true`	`Group is false`	`Group is false`

Both criteria are similar - they retrieve a value and compare it to something:

The first criteria is fairly simple:

- Field to Compare. We indicate that the value of the Low Income Indicator characteristic type kept on the main person linked to the account should be retrieved. We've assumed that this characteristic type has values of `YES` and `NO`.
- Comparison Method. We indicate that if the person has a characteristic value of `YES`, then we should check the next condition; otherwise we indicate the group is false (and if the group is false, the rate component is skipped).

The second criterion works similarly:

- Field to Compare. We indicate the service quantity (SQ) that holds the total consumed quantity of a product should be retrieved.
- Comparison Method. We simply compare the value to 1,000.

If the second criterion is true, then the group is true and the rate component will be applied; otherwise, the group is false and the rate component will be skipped.

A Rate Component With A Time Span Comparison

A rate component that is only eligible for senior citizens has the following eligibility rule:

- Birth date equates to that of a senior citizen

This rule requires only one eligibility group on the rate component.

Copyright © 2024, Oracle and/or its affiliates.

The following table illustrates the eligibility group defined on the rate component:

Group No.	Group Description	If Group is TRUE	If Group is FALSE
1	Customer Is A Senior Citizen	`Apply rate component`	`Skip rate component`

The following table illustrates the criteria defined in the eligibility group:

Group 1: Banking, Calif, Senior					
Seq	**Field to Compare**	**Comparison Method**	**If TRUE**	**If FALSE**	**If Insufficient Data**
10	Main person characteristic: Date of Birth	Algorithm: True if senior	`Group is true`	`Group is false`	`Group is false`

The criterion contains a time span comparison. Time span comparisons are used to compare a date to something. In our example, we have to determine the age of the customer based on their birth date. If the resultant age is > 65, they are considered to be a senior citizen. To pull this off, you can take advantage of a comparison algorithm supplied with the base rate component as described below.

- Field to Compare. The person characteristic in which the customer's birth date is held is selected.
- Comparison Method. We chose a comparison algorithm that returns a value of `TRUE` if the related field value (the customer's date of birth) is greater than 65 years (refer to RECC-TIMESPN for an example of this type of algorithm).

You'll notice that if a value of `TRUE` is returned by the `True if senior` algorithm, the group is true (and we've setup the group to indicate a true group means the rate component is eligible).

Note: The time span algorithm can be used to compare days, weeks, months, etc. Refer to RECC-TIMESPN for more information about this algorithm.

A Rate Component With Service Type Comparison

Imagine a rate component that is only eligible if the current customer has banking service AND insurance service. This rate component would need the following eligibility rules:

- Customer has banking service, and
- Customer has insurance service

These rules require only one eligibility group on the rate component.

The following table illustrates the eligibility group defined on the rate component:

Group No.	Group Description	If Group is TRUE	If Group is FALSE
1	Has banking service AND insurance service	`Apply rate component`	`Skip rate component`

The following table illustrates the criteria defined in the eligibility group:

Group 1: Has banking service AND insurance service					
Seq	**Field to Compare**	**Comparison Method**	**If TRUE**	**If FALSE**	**If Insufficient Data**
10	Algorithm: check if customer has banking service	= TRUE	`Check next condition`	`Group is false`	`Group is false`

Copyright © 2024, Oracle and/or its affiliates.

Group 1: Has banking service AND insurance service					
Seq	**Field to Compare**	**Comparison Method**	**If TRUE**	**If FALSE**	**If Insufficient Data**
20	Algorithm: check if customer has insurance service	= TRUE	`Group is true`	`Group is false`	`Group is false`

Both criteria are similar - they call an algorithm that performs a logical comparison. These algorithms are a bit counter intuitive (but understanding them will provide you with another way to implement complex eligibility criteria):

Both criterion works as follows:

- Field to Compare. We chose a "field to compare" algorithm that checks if the current account has contracts that belong to a given set of service types. It returns a value of `TRUE` if the customer has an active contract that matches one of the service types in the algorithm. In our example, the "check if customer has banking service" algorithm returns a value of `TRUE` if the customer has at least one active contract whose contract type references the banking service type. The "check if customer has insurance service" algorithm is almost identical, only the service type differs. Refer to RECF-SRVTY and RECF-AUTOPAY for examples of this type of algorithm.
- Comparison Method. We simply compare the value returned by the algorithm to TRUE and indicate the appropriate response.

Note: Bottom line. The "field to compare" algorithm isn't actually returning a specific field's value. Rather, it's returning a value of `TRUE` or `FALS` (this is not a misspelling). This value is, in turn, compared by the "comparison method" and the group is set to true, false or check next accordingly.

Defining Rate Components

CAUTION: We strongly recommend that you familiarize yourself with the information under Designing Rate Components before using this transaction.

Rate components control exactly how your bill calculation lines are calculated. Every rate component exist in respect of a given *rate version*. This is because the rate version defines the date on which all of its rate components become effective.

Note: Copying, delete and moving rate components. You can use the rate component maintenance transaction to duplicate, delete and move rate components. However, you may find these functions are easier to do using the Rate Version Merge transaction as this transaction can copy en masse and it also updates the relevant cross reference information if you move or delete a rate component.

The topics in this section describe how to maintain a rate component.

Note: Don't forget to finish the rate version. After you have added all necessary rate components to a rate version, don't forget to return to Rate Version - Main and change the state of the rate version to `Finished` (otherwise, it cannot be used by billing).

Rate Component - Main Information

Core information about a rate component is defined using **Main Menu, Rates, Rate Component**.

Description of Page

Rate Component contains a concatenation of basic information about the rate component. This information only appears after the rate component has been added to the database. The adjacent up and down arrows cause the rate component immediately before or after this rate component to be displayed.

Copyright © 2024, Oracle and/or its affiliates.

Rate Version defines the rate version to which the rate component is linked. **Sequence** defines the relative position of this rate component in respect of the other rate components. The position is important because it defines the order in which the system calculates the charges AND the order in which the calculated charges appear on bills. **Rate Version** and **Sequence** form the unique identifier of the rate component. Both fields become protected after the rate component is added to the database.

Note: Re-sequencing rate components. You must use the Rate Version Merge transaction to reposition a rate component. If you use the up / down arrows on the Rate Version Merge, the system will change the rate component's **Sequence** and change all rate components that reference the old **Sequence** accordingly.

Note: Leave gaps in the sequence numbers. Make sure you leave space between sequence numbers so that you can add new rate components between existing ones in the future. If you run out of space between rate components, you can use the Rate Version Merge transaction to re-sequence a rate version's rate components.

Note: Eligibility criteria are highlighted. If the rate component has eligibility criteria, an indication of such appears.

Select the **Rate Component Type** that corresponds with the rate component. The screen will display the appropriate fields, which correspond with the categories of rate components described in Designing Rate Components. This field will be gray when the rate component is referenced on other rate components.

CAUTION: The Rate Component Type affects what you can enter on other parts of the window. The remainder of this section is devoted to those fields that can be entered regardless of RC Type. The subtopics that follow describe those fields whose entry is contingent on the RC Type.

Use **Description** to describe what the rate component does.

Rounding Type and **Precision** control how the system rounds the rate component's calculated value. Refer to Rate Component Rounding for a complete description.

Note: Rounding Type defaults to `Nearest` and **Precision** defaults to a value consistent with the decimal positions defined on the rate schedule's currency code.

The **Allow Currency Conversion** field indicates whether the currency conversion feature must be enabled or disabled for the rate component. If the **Allow Currency Conversion** field is set to **Yes**, the currency conversion is done for the rate (i.e. rate component) based billable charges. By default, the value is set to **No**. The **Allow Currency Conversion** field is enabled when the **Allow Multi-Currency Accounts** option type of the **Multi-Currency Accounts (C1_MLTCURACC)** feature configuration is set to **Y**. Otherwise, the **Allow Currency Conversion** field is disabled.

Turn on **For Calculation Purposes Only** (FCPO) if this rate component exists purely to calculate the percent / flat rate / unit rate used by another rate component. When this switch is on, the system does not include the calculated amount in the bill total.

When this switch is on:

- You must define what is calculated in **Result Type**; permissible values are `Unit Rate`, `Charge`, `Step Multiplier` and `Percentage`.
- Use **Create a Bill Line** to define if a bill line should be created for the rate component (if you don't need to show the results to the customer or a user, there is no need to create a bill line).
- If a bill line is being created, use the **FCPO Retention Rule** to define if the FCPO amount should be set to zero on the bill line or whether the bill line's amount should be set equal to the FCPO amount.

Note: If the rate component is an FCPO rate component, the system uses the rounding type and precision defined on the rate component. And, if the rate component is a non-FCPO rate component, the system uses decimal positions specified for the currency (in the **Currency** screen) and rounding type defined for the currency (in the **Division** screen).

For examples of rates that use the FCPO functionality, refer to Calculated Minimum Charges and Step Boundaries Based On The Number Of Days Of Usage And The Region In Which The Customer Resides.

Copyright © 2024, Oracle and/or its affiliates.

For each rate component, you will need to select the **Value Type** and **Value Source**. Valid values for the Value Type are `Charge`, `Percentage` and `Unit Rate`. More information to help determine which value type to choose are described in the How To sections below. Based on the Value Source entered, the remainder of this row will change.

If the Value Source is `Bill Factor`, then a prompt to indicate the **Bill Factor** will appear.

If the Value Source is `Other Rate Component`, then a prompt to indicate the **RC Sequence** will appear.

If the Value Source is `Value`, then a prompt to indicate the **Value** directly on the rate component will appear.

If the Value Source is `Value Algorithm`, then a prompt to indicate the **Value Algorithm** will appear.

If you plan to use this method, you must set up this algorithm in the system. This can be done with the following options:

- Create a new algorithm (refer to *Setting Up Algorithms*).
- On this algorithm, reference an Algorithm Type that calculates a rate component's value. Click here to see the algorithm types available for this plug-in spot.

Fastpath: For more information about which method to use, refer to Designing Rate Components.

If the charges associated with the rate component are only levied during a specific season, you must turn on the **Seasonal** options.

When **Seasonal** is turned on, you use **Prorate Method** to define how the system should prorate a seasonal charge when a bill segment's start and end dates are not entirely within the seasonal period (defined in **Season**). The following options are available:

- `Accounting Date Dependent.` This option will not prorate the charge calculated by this rate component. Rather, it will only calculate the charge if the accounting date associated with the bill segment is within the seasonal period.
- `Bill End Date.` This option will not prorate the charge calculated by this rate component. Rather, it will only calculate the charge if the bill segment end date is within the seasonal period.
- `Bill Start Date.` This option will not prorate the charge calculated by this rate component. Rather, it will only calculate the charge if the bill segment start date is within the seasonal period.
- `Prorate.` This option will prorate the charge calculated by this rate component based on the number of days in the adjacent season that is included in the bill period. For example, if a bill period is from 15-April through 15-May and the seasonal period is from 1-May through 31-October, the system will assume 15 / 30 of the charge should be levied.
- `Prorate seasonal SQ.` This option is only pertinent if you have:
  - Seasonal service quantity (SQ) charges, AND
  - A bill that crosses seasonal boundaries, AND
  - Multiple rate versions effective across the bill period.
- If this option is used, the system will prorate the charge calculated by a SQ rate component based on the number of days in the adjacent season that is included in the rate version's calculation period.

Note: The **Season** al periodis defined in the two adjacent fields. The first field contains the day and month when the season starts; the second field contains the day and month when the season ends. The day and month should be entered in the format defined in your display profile.

Description On Bill, **Print**, and **Print If Zero** all control if the rate component contributes a line to the customer's bill. These fields can also be modified on Rate Version - Bill Print Info. You might find it easier to setup these fields on the rate version transaction as you can copy and paste descriptions between rate components. Refer to Rate Version - Bill Print Info for a description of these fields.

The other fields on this page are dependent on the type of rate component. See the "how to" subtopics below for more information.

How To

Copyright © 2024, Oracle and/or its affiliates.

How To Set Up Flat Charge Rate Components

There are no special additional fields on this page available for setting up a Flat rate component. The following information will help you determine how to set up your Flat rate components.

The **Value Type** in this case would be `Charge`. This field will be gray when the rate component is referenced on another rate component.

Refer to Rate Component - Main Information for information about defining the Value Source.

Enter the verbiage to appear on the customer's bill in **Description On Bill** and turn on the **Print** switch. Refer to How to use Description on Bill for more information about these fields.

Move to the **GL Distribution** window to define how to book moneys associated with this rate component in the general ledger.

How To Set Up Service Quantity Rate Components

When setting up an SQ rate component, additional fields become available for you to define.

The following information will help you to set up your SQ rate components.

Select a **Value Type** of `Unit Rate` if the type of charge is an amount per some unit of product/charge type. In the rare situation where the amount of the charge is flat for the first X units of product/charge type, use the Value Type of `Charge`. This field will be gray when the rate component is referenced on another rate component.

Refer to Rate Component - Main Information for information about defining the Value Source.

The amount that you specified using the bill factor / value field was a price per some unit of some thing. This thing must be identified using the **Unit Of Measure**, **Variance Parameter**, and **Service Qty. Identifier** fields.

Fastpath: For more information about UOM / Variance Parameter/ SQI, refer to UOM versus Variance Parameter versus SQI.

Turn on **Error if No Value** if a bill error should be generated if the UOM / Variance Parameter/ SQI specified on the rate component was not supplied at billing time.

If the charge is only applicable to product quantity consumption within some tier which indicates that the rate component is **Stepped**. When this option is turned on, you must indicate the **Step Low Value** and **Step High Value** that the product quantity consumption charge is applicable to. For example, if the product price is applicable to the first 300, then you would enter **Step Low Value**0 and **Step High Value**300. When multiple tiers exist, the high value of the first tier is the low value of the second tier (and so on). To extend our example, the second tier would have a low value of 300; the high value may be left blank because the system assumes a blank high value means "to infinity".

If the step boundaries are dynamic (i.e., they are calculated based on something that's only known at billing time), you can change the step boundaries using either of the following methods:

- Use **Step RC Sequence** to indicate the sequence number of the rate component whose value will be multiplied by the step's boundaries to calculate the step boundaries to use at billing time.
- Use **Step Algorithm** to indicate that an algorithm will be called to manipulate the step boundaries (note, this algorithm can manipulate the low and/or high boundaries). If you plan to use this method, you must set up this algorithm in the system. To do this:
  - Create a new algorithm (refer to *Setting Up Algorithms*).
  - On this algorithm, reference an Algorithm Type that manipulates a rate component's step boundaries. Click here to see the algorithm types available for this plug-in spot.

Fastpath: For an example of when dynamic steps are needed, refer to Step Boundaries Based On The Number Of Days Of Usage And The Region In Which The Customer Resides.

CAUTION: If you dynamically calculate step boundaries and the resultant high AND low values become zero, this signals the system that the rate component should be skipped (i.e., no bill line will be produced for the rate component).

Copyright © 2024, Oracle and/or its affiliates.

Turn on **GL Statistical Quantity** if GL journal lines generated for this rate component should also contain the service quantity amount as a statistical quantity. You would use this option if you keep track of both dollar amounts and consumption units in your general ledger.

CAUTION: When a bill segment is created, the system stores the statistical quantity on the journal line associated with the rate component's distribution code. If you book multiple UOM's to the same distribution code, you should only turn on GL Statistical Qty on one of the rate components. Otherwise, you will commingle different UOM's on a journal line's statistical quantity.

Fastpath: Refer to Determining the Statistical Quantity for more information.

Turn on **Measures Peak Qty** if the UOM / Variance Parameter/ SQI is associated with banking or any other related domain.

Note: The value of **Measures Peak Qty** is defaulted from the **UOM** table (if a UOM is specified).

Enter the verbiage to appear on the customer's bill in **Description On Bill** and turn on the **Print** switch. Refer to How to use Description on Bill for more information about these fields.

Move to the **GL Distribution** window to define how to book moneys associated with this rate component in the general ledger.

How To Set Up Apply To Rate Components

There are no special additional fields on this page available for setting up an Apply To rate component. The following information will help you determine how to set up your Apply To rate components.

Select a **Value Type** of `Percent` if the type of charge is a percent of the dollar value of previously calculated rate components. If the amount being charged is an amount per some unit of product/charge type, select the Value Type of `Unit Rate`. This field will be gray when the rate component references other rate components.

Refer to Rate Component - Main Information for information about defining the Value Source.

Enter the verbiage to appear on the customer's bill in **Description On Bill** and turn on the **Print** switch. Refer to How to use Description on Bill for more information about these fields.

Move to the **Cross Reference** window to define the rate components to which the charge will be applied.

Move to the **GL Distribution** window to define how to book moneys associated with this rate component in the general ledger.

How To Set Up Maximum Charge Rate Components

There are no special additional fields on this page available for setting up a Maximum rate component. The following information will help you determine how to set up your Maximum rate components.

Select a **Value Type** of `Charge`.

Refer to Rate Component - Main Information for information about defining the Value Source.

Enter the verbiage to appear on the customer's bill in **Description On Bill** and turn on the **Print** switch. Refer to How to use Description on Bill for more information about these fields.

Move to the **Cross Reference** page to define the rate components against which the maximum charge will be applied.

Move to the **GL Distribution** page to define how any additional revenue associated with the maximum charge should be booked in the general ledger.

How To Set Up Minimum Charge Rate Components

There are no special additional fields on this page available for setting up a Minimum rate component. The following information will help you determine how to set up your Minimum rate components.

Select a **Value Type** of `Charge`.

Copyright © 2024, Oracle and/or its affiliates.

Refer to Rate Component - Main Information for information about defining the Value Source.

Enter the verbiage to appear on the customer's bill in **Description On Bill** and turn on the **Print** switch. Refer to How to use Description on Bill for more information about these fields.

Move to the **Cross Reference** window to define the rate components against which the minimum charge will be applied.

Move to the **GL Distribution** window to define how any additional revenue associated with the minimum charge should be booked in the general ledger.

How To Set Up Exact Charge Rate Components

There are no special additional fields on this page available for setting up an Exact Charge rate component. The following information will help you determine how to set up your Exact Charge rate components.

Select a **Value Type** of `Charge`.

Refer to Rate Component - Main Information for information about defining the Value Source.

Enter the verbiage to appear on the customer's bill in **Description On Bill** and turn on the **Print** switch. Refer to How to use Description on Bill for more information about these fields.

Move to the **Cross Reference** window to define the rate components against which the exact charge will be applied.

Move to the **GL Distribution** window to define how any additional revenue associated with the minimum charge should be booked in the general ledger.

How To Set Up Summary Rate Components

Summary rate components use no other fields on this page other than those defined above.

Enter the verbiage to appear on the customer's bill in **Description On Bill** and turn on the **Print** switch. Refer to How to use Description on Bill for more information about these fields.

Move to the **Cross Reference** window to define the rate components to summarize on this rate component.

How To Set Up Calculation Algorithm Rate Components

Calculation Algorithm rate components are used when none of the other rate components will provide you with the functionality you need.

First, you must select a **Calc Algorithm**. This calculation algorithm will have the responsibility of producing your bill lines. All rate component options become available when you use this rate component type, based on the assumption that it is generally capable of doing anything you might require.

If you plan to use this method, you must set up this algorithm in the system. This can be done with the following options:

- Create a new algorithm (refer to *Setting Up Algorithms*).
- On this algorithm, reference an Algorithm Type that calculates a rate component. Click here to see the algorithm types available for this plug-in spot.

Note: The calculation algorithm's main purpose is to create bill calculation lines. However, the algorithm may populate other information for the bill, for example, it may add to the SQ or register read collection or it may overwrite the description on bill.

If you wish to provide your users with the ability to audit the bill lines created by a rate component of this type, you may indicate an **Audit Alg** orithm.

Fastpath: Refer to *Setting Up Interval Pricing Rate Components* and *Setting Up Variance Parameter Pricing Rate Components* for more information about the audit algorithms provided with the system.

All the other fields on the page are optional fields available to you for use by your calculation algorithm.

Turn on **Error if No Value** if a bill error should be generated if the UOM / Variance Parameter/ SQI specified on the rate component was not supplied at billing time.

Copyright © 2024, Oracle and/or its affiliates.

Turn on the **Derive SQ** switch when your calculation algorithm will be creating entries in the SQ array for the UOM / Variance Parameter/ SQI defined on your rate component. In this case, the system will not attempt to locate an entry in the SQ collection matching the rate component's UOM / Variance Parameter/ SQI before calling your calculation algorithm.

Enter the verbiage to appear on the customer's bill in **Description On Bill** and turn on the **Print** switch. Refer to How to use Description on Bill for more information about these fields.

Move to the **GL Distribution** window to define how any additional revenue associated with the minimum charge should be booked in the general ledger.

Rate Component - Cross Reference

Minimum charge, maximum charge, exact charge, apply-to, and summary rate components reference other rate components. For these types of rate components, open **Main Menu, Rates, Rate Component** and navigate to the **Cross Reference** tab define the other rate components to which it refers.

Description of Page

Rate Component contains a concatenation of basic information about the rate component. This information only appears after the rate component has been added to the database. The adjacent up and down arrows cause the rate component immediately before or after this rate component to be displayed.

Use the **Apply this Rate Component to** collection to specify the other rate component(s) on the Rate Version to which the rate component in question should refer. The list of rate components available for reference is displayed on the right. Use the left arrow to move the desired rate components into the grid. If a rate component appears in the apply to list in error, use the left arrow to remove the rate component.

Fastpath: Refer to Editable Grid in the system wide standards documentation for more information about adding records to a collection by selecting from a list.

Note: An apply to rate component that has a charge type of `unit rate` can only reference SQ, item, interval pricing and calculation rate components. Apply to rate components with a charge type of `percentage`, along with summary, minimum charge, maximum charge and exact charge rate components can reference all types of rate components including summary.

The **This RC is referenced by** collection indicates what other rate components reference the rate component in question.

Rate Component - GL Distribution

Open **Main Menu, Rates, Rate Component** and navigate to the **GL Distribution** tab to define each rate component's effect on the general ledger. GL distribution information is required for all rate components except for FCPO and summary rate components.

Description of Page

Rate Component contains a concatenation of basic information about the rate component. This information only appears after the rate component has been added to the database. The adjacent up and down arrows cause the rate component immediately before or after this rate component to be displayed.

If the rate component's charges should be booked to a single GL account regardless of the contract's contract type's Revenue Class, enter a **Distribution Code**. If the rate component's charges are booked to a different GL account depending on the contract's Revenue Class, turn on **Use Revenue Class** and insert the appropriate **Revenue Class** and **Distribution Code** for each that uses the rate.

Copyright © 2024, Oracle and/or its affiliates.

Rate Component - Characteristics

Your algorithms and reports may need additional rate component fields that aren't supported in the base-package. If so, you can set up a rate component characteristic for each such field.

Note: Snapshotting characteristics on bill calculation lines. The system will automatically copy a rate component's characteristics onto the resultant bill lines for those characteristic types that are usable on bill lines.

Open **Main Menu, Rates, Rate Component** and navigate to the **Characteristics** tab to define characteristic values for your rate component.

Description of Page

For each characteristic in the collection, indicate the appropriate **Characteristic Type** and **Characteristic Value**.

Note: You can only choose characteristic types defined as permissible on the rate schedule record. Refer to Setting Up Characteristic Types & Their Values for more information.

Rate Component - Eligibility

This page is used to define the conditions under which a rate component will be applied. If you don't specify any **Eligibility Criteria Groups**, the system assumes the rate component should be applied (i.e., a rate component is eligible by default).

Fastpath: Refer to The Big Picture Of Rate Component Eligibility for more information.

Open this page using **Main Menu, Rates, Rate Component** and navigate to the **Eligibility** tab.

Description of Page

Rate Component contains a concatenation of basic information about the rate component. This information only appears after the rate component has been added to the database. The adjacent up and down arrows cause the rate component immediately before or after this rate component to be displayed.

CAUTION: The following information is not intuitive; we strongly recommend that you follow the guidelines under The Big Picture Of Rate Component Eligibility before attempting to enter this information.

The **Eligibility Criteria Group** scroll contains one entry for each group of eligibility criteria. The following fields may be defined for each group:

- Use **Sort Sequence** to control the relative order in which the group is executed when the system determines if the rate component should be applied (smaller numbers are executed before larger numbers).
- Use **Description** and **Long Description** to describe the criteria group.
- Use **If Group is True** to define what should happen if the eligibility criteria (defined in the following grid) return a value of `True`.
  - Choose `Apply Rate Component` if this rate component should be executed.
  - Choose `Check Next Group` if the next criteria group should be checked.
  - Choose `Skip Rate Component` if this rate component should NOT be executed.
- Use **If Group is False** to define what should happen if the eligibility criteria (defined in the following grid) return a value of `False`.
  - Choose `Apply Rate Component` if this rate component should be executed.
  - Choose `Check Next Group` if the next criteria group should be checked.
  - Choose `Skip Rate Component` if this rate component should NOT be executed.

The grid that follows contains the rate component's eligibility criteria. Think of each row as an "if statement" that can result in the related eligibility group being True or False. For example, you might have a row that indicates the customer

Copyright © 2024, Oracle and/or its affiliates.

is eligible for the rate component if the customer's birth date equates to that of a senior citizen. The following bullets provide a brief description of each field on an eligibility criterion. Please refer to Defining Logical Criteria for several examples of how this information can be used.

- Use **Sort Sequence** to control the order in which the criteria are checked.
- Use **Criteria Field** to define the field to compare:
  - Choose `Characteristic` if you want to compare a characteristic value that resides on any of the following to a given value (use the adjacent fields to define the object on which the characteristic resides and the characteristic type):
- The contract being billed
- The contract's account
- The main person linked to the contract's account
- The price item (product)
- The price assignment
- In addition, you can also use a characteristic value that is derived while the rate is being calculated (characteristic values can be created by SQ rules, register rules and many other rate component algorithms)
  - Choose `Rate Component Value` if you want to compare the final value of an earlier rate component to a given value. Push the adjacent search button to select the rate component's sequence number.
  - Choose `Service Quantity` if you want to compare the final value of a service quantity to a given value. Push the adjacent search button to select the service quantity's **UOM/ Variance Parameter/ SQI**.
  - Choose `Algorithm` if you want to compare anything other than the above options. Push the adjacent search button to select the algorithm that is responsible for retrieving the comparison value.
  - Use **Criteria Comparison** to define the method of comparison:
    - Choose `Algorithm` if you want an algorithm to perform the comparison and return a value of True, False or Insufficient Data.
    - Choose any other option if you want to compare the **Criteria Field** using a logical operator. The following options are available:
  - Use >, <, =, >=, <=, <> (not equal) to compare the **Criteria Field** using standard logical operators. Enter the comparison value in the adjacent field.
  - Use `IN` to compare the **Criteria Field** to a list of values. Each value is separated by a comma. For example, if a field value must equal `1`, `3` or `9`, you would enter a comparison value of `1,3,9`.
  - Use `BETWEEN` to compare the **Criteria Field** to a range of values. For example, if a field value must be between `1` and `9`, you would enter a comparison value of `1,9`. Note, the comparison is inclusive of the low and high values.
  - The next three fields control whether the related logical criteria cause the eligibility group to be considered True or False:
    - Use **If TRUE** to control what happens if the related logical criterion returns a value of True. You have the options of `Group is true`, `Group is false`, or `Check next condition`. If you indicate `Group is true` or `Group is false`, then the rate component will be `Skipped` or `Applied` based on the values defined above in **If Group is False** and **If Group is True**.
    - Use **If FALSE** to control what happens if the related logical criterion returns a value of False. You have the options of `Group is true`, `Group is false`, or `Check next condition`. If you indicate `Group is true` or `Group is false`, then the rate component will be `Skipped` or `Applied` based on the values defined above in **If Group is False** and **If Group is True**.
    - Use **If INSUFFICIENT DATA** to control what happens if the related logical criterion returns a value of "Insufficient Data". You have the options of `Group is true`, `Group is false`, or `Check next condition`. If you indicate `Group is true` or `Group is false`, then the rate component will be `Skipped` or `Applied` based on the values defined above in **If Group is False** and **If Group is True**.

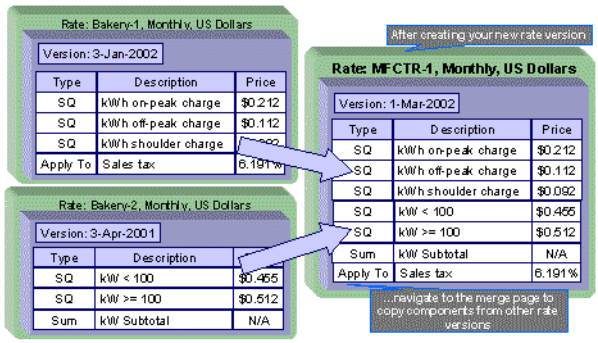
Rate Version Merge

The following point summarize the many diverse functions available on this transaction:

Copyright © 2024, Oracle and/or its affiliates.

- A rate version's rate components have unique sequence numbers. This number controls the order in which a rate component is processed. If you need to insert a rate component and there's no space between existing sequence numbers, you can use this transaction to renumber the rate version's rate components.
- You can use this transaction to move a rate component to a different position within a rate version. When a rate version is moved, all references to the rate component are changed to reflect the new sequence number.
- You can use this transaction to delete a rate component. When a rate component is deleted, all references to the rate component are deleted.
- You can use this transaction to copy rate components from other rate versions. For example,
  - You may want to create a rate version that is similar to an existing rate version. Rather than copying all the information from the existing rate version and then removing the inapplicable components, this page may be used to selectively copy specific rate components from the existing rate version to the new rate version.
  - You may have customer-specific rates that are very similar, but still unique. You can use this transaction to build a customer-specific rate. In this scenario, you may choose to create special 'mini' rate versions, one for each of the various options. Then, you could use the rate version merge page to select the components applicable for the new custom rate version.

Note: The target rate version must exist prior to using this page. If you are creating a new rate, you must first create the Rate Schedule and Rate Version and then navigate to the merge page to copy rate component information.



Note: Duplicate versus Merge. The Rate Version page itself has duplication capability. You would duplicate a rate version if you want to a) create a new rate version AND b) populate it with *all* the rate components from an existing rate version. The Rate Component Maintenance transaction also allows you to duplicate a rate component within a rate version.

Open **Main Menu, Rates, Rate Version Merge** to open this page.

Description of Page

Select the **Original Rate Version**, which is the target for merging the rate component information.

Note: Finished rate versions are protected. Only rate versions that are `In Progress` or `Validated` may be used as the target rate version in the merge functionality. If any changes are made to a `Validated` rate version on this page, the system will move the status back to `In Progress`. If a rate version with a status of `Finished` is selected, it will be displayed, but you will be unable to copy any information to it (and its description will appear in red). If you need to make changes to a `Finished` rate version, navigate to Rate Version - Main and change the rate version's status to `In Progress`. Refer to Lifecycle of a Rate Version for more information.

Select the **Merge From Rate Version**, which is your template rate version to copy the rate components from.

Note: You may only copy components from one Merge From rate version at a time. If you want to copy components from more than one rate version, select the first Merge From rate version, copy the desired records, Save, and then select the next Merge From rate version.

The left portion of the page displays any existing rate components for the **Original Rate Version**. The right portion of the page displays the existing rate components for the **Merge From Rate Version**.

Copyright © 2024, Oracle and/or its affiliates.

You can use the **Copy All** button to copy all the rate components from the **Merge From** rate to the **Original** rate. If you use **Copy All**, please be aware that the rate components are added to the end of the **Original** rate version.

Each time you save the changes, the system renumbers the **Original** rate version using the **Start From Sequence Number** and **Increment By**.

The left portion of the page initially displays existing rate component records linked to the original rate version. In the **Merge Type**, you will see the word `Original`, for any of these records. The **Sequence**, **RC Type** and **Description** for each rate component are displayed. In the right portion of the page, the existing records in the merge from rate version are displayed initially.

The topics that follow describe how to perform common maintenance tasks:

Resequencing Rate Components

If there are no gaps between two rate components and you need to resequence the rate components:

- Make a change to the **Start From Sequence Number** or **Increment By** (so that the Save button becomes enabled).
- Click the Save button.

Adding A New Row To A Rate

You may move any of the rate components from the Merge From rate version to the original rate version by selecting the left arrow adjacent to the desired row. Once a record is moved it will disappear from the Merge From information and appear in the Original information with the word `Merge` in the Merge Type column.

Removing An Uncommitted Row From A Rate

If you have copied a row across by mistake, you may remove it by clicking on the right arrow adjacent to the appropriate record.

Moving Positional Rows Up and Down

A rate component's sequence number controls the order in which the rate component should be applied. You may modify the order execution using the up and down arrows.

The sequences need to be renumbered in order to ensure the correct order. The fields **Start from sequence number** and **Increment By** are used to define the sequence number to assign to the first rate component and the subsequent values to assign to subsequent rate components.

Fastpath: Refer to Editable Grid in the system wide standards documentation for more information about adding records to a collection by selecting from a list and repositioning rows within a grid.

Rate Component Cross Reference Information

All information related to a rate component, including general ledger information, characteristics and eligibility rules, will be copied.

Special logic exists for merged rate components that reference other rate components. This includes the cross-reference collection and the eligibility criteria.

This logic is difficult to explain succinctly, so we'll use an example. Imagine that the Merge From rate components look as follows:

Copyright © 2024, Oracle and/or its affiliates.

Seq	RC Type	Description
10	Service Quantity	0 to 300 kWh step
20	Service Quantity	300 to max kWh
30	Summary	Summary of stepped charges
32	Service Quantity	Days Calculation Rate
34	Minimum Charge	Minumum Charge
100	Apply-to	Employee discount (rider req)
200	Apply-to	City tax
210	Apply-to	State tax

The Rate Component - Cross Reference tab for the merge from rate version shows the following cross-reference information for sequence `210`:

Apply This Rate Component To:

RC Sequence	Description
10	0 to 300 kWh step
20	300 to max kWh
34	Minumum Charge
100	Employee discount (rider req)

Assume that only rate components 10, 20 and 210 are copied to the new rate version and assume that the Original rate version already contains a rate component with sequence number 10 that is moved to the second position.

Start from sequence number 10 **Increment By** 10

Merge Type	Sequence	RC Type	Description
Merge	10	Service Quantity	0 to 300 kWh step
Original	10	Flat Charge	Customer Service Charge
Merge	20	Service Quantity	300 to max kWh
Merge	210	Apply-to	State tax

After saving, the second service quantity rate component is renumbered and the State tax rate component will reflect the renumbering in the cross-reference and will not include the rate components that were not copied.

Here is the view on the merge page after saving:

Start from sequence number 10 **Increment By** 10

Merge Type	Sequence	RC Type	Description
Original	10	Service Quantity	0 to 300 kWh step
Original	20	Flat Charge	Customer Service Charge
Original	30	Service Quantity	300 to max kWh
Original	40	Apply-to	State tax

If you navigate to the Rate Component - Cross Reference tab for the original rate version for sequence `40- State tax`, it will look as follows:

Apply This Rate Component To:

Sequence	Description
10	0 to 300 kWh step
30	300 to max kWh

If any rate component refers to another rate component in the eligibility criteria, the sequence number of the eligibility criteria is also adjusted accordingly.

Copyright © 2024, Oracle and/or its affiliates.

> **Note:** You will be warned by the system when you copy a rate component that has cross-references to other rate components that are not being copied.

Rate Check

The Rate Check page allows you to test your rates under various scenarios. The topics in this section describe how to use this transaction.

Rate Check - Main

Open **Main Menu, Rates, Rate Check** to enter the parameters required to check a rate.

Description of Page

> **Note: Easiest way to use rate check.** The easiest way to populate the above parameters is to select an existing bill segment on Bill Segment - Calc Lines and then use the rate schedule context menu to navigate to this transaction. When you do this, the system populates the above fields from the values on the bill segment. After these fields are populated, you can override the defaulted values to check various scenarios.

Indicate the **Rate Schedule** that you want to test. The version(s) used will be those in effect during the start and end dates. Only `validated` and `finished` rate versions will be used.

> **Note:** If **Rate Schedule** is blank and you enter a **Contract ID**, the contract's current rate will be defaulted.

While using a **Contract ID** is not strictly required to check a rate, we strongly recommend finding a representative contract to check a rate. Why? Because charges in a rate may be affected by values in a contract. For example, consider the following types of rate components that use contract-specific values:

- Rate components with bill factors that are subject to the presence of contract riders. In fact, if a bill factor requires a contract rider and you do not specify a **Contract ID** rate check assumes that the rider is applicable.
- Rate components with bill factors that are subject to tax exemptions.
- Rate components with bill factors with customer-specific prices.
- Rate components with service quantity (SQ) rules that use characteristics on contract and account.
- Rate components with SQ rules that use contract quantities.

In other words, if you don't specify a **Contract ID**, you will not be able to check rates with the above characteristics.

Enter a **Bill Seg ID** if you want to test the rate with actual usage for a given bill segment. This is useful if you want to compare the actual bill segment results with the results of the same usage under a different rate schedule. If you indicate a **Bill Seg ID**, rate check's **Contract ID** will be populated with the bill segment's contract ID (and **Contract ID** will be protected).

> **Note: Overriding product/charge type.** If you enter a **Bill Seg ID**, the bill segment's product/charge type will be defaulted into the product/charge type array that follows. You may override this product/charge type to test various scenarios.

You must indicate a **Revenue Class**. This is needed for applying the appropriate GL distribution code for rate components where the GL account is based on the customer's revenue class.

Enter the **Start Date** and **End Date** that you are testing. If a Bill Seg ID is entered, the start and end dates are taken from the bill segment's period, but can be overwritten.

Indicate the **Accounting Date**. If a **Bill Seg ID** is entered, the accounting date on the bill segment is used but can be overwritten.

Enter the quantities that you want to test in the quantity collection. For every **UOM**/ **Variance Parameter**/ **SQI** combination, enter a register quantity.

Copyright © 2024, Oracle and/or its affiliates.

The collection of **Characteristic Types** and **Characteristic Values** is used to apply the appropriate bill factor values. If these values are entered, these values will be used to apply the appropriate bill factors (even if you have specified a contract from which characteristics can be derived). If these values are blank AND there is no contract ID, then you will receive an error from the rate application indicating which values are needed.

Indicate the **Bill Language** to use for the resulting bill calc line descriptions.

When you have entered everything, click on the **Check Rate** button. This will cause your rate to be applied to the parameters entered above. If the system is successful, you will be transferred to the **Calc Lines** page on which the resulting calculation details can be viewed.

Rate Check Results - Calc Lines

After you have pressed the **Check Rate** button on the **Main** page, you will be transferred to the **Calc Lines** tab to view the lines that have been calculated by your rate.

Description of Page

The **Rate Schedule** and its description, the **Revenue Class** and its description and the **Bill Period** are displayed at the top.

More than one Rate Version may have been effective for the Rate Schedule during this bill period. If so, multiple collections of calculation lines will be produced. Use the **Bill Seg Hdr** scroll to view each collection of calculation lines.

For each Bill Seg Hdr, the following information will be displayed above the calculation line grid.

Sequence This identifies which Bill Seg Hdr is being displayed.

Start Date, End Date This is the period, within the bill period, that the rate version, which produced these lines, was effective.

Amount This is the total amount calculated by this group of calculation lines.

Desc on Bill The description on bill for the rate version that produced these lines.

Rate Version The rate version that was used to produce these lines.

The grid, which displays the bill calculation lines for this rate version, shows one row exists for every calculation involved in this process. This information is very similar to what is displayed on the bill calculation lines page for a bill segment. The following information is displayed in the grid:

- **Sequence** is the system-assigned unique identifier of the calculation detail row.
- **Description on Bill** is the information about the bill line that appears on the customer's bill.
- **Calculated Amount** is the calculated amount associated with the bill line.
- The **Print** switch controls whether information about this line will print on the customer's bill.
- The **App in Summary** switch defines if this line's amount also appears on a summary line. This switch is turned on if the corresponding rate component is summarized on a summary rate component.
- **UOM** is the unit of measure of the service quantity priced on the calculation line.
- **Sub-UOM** is the Sub-Unit of Measure code of the service quantity priced on the calculation line.
- **SQI** is the service quantity identifier of the service quantity priced on the calculation line.
- **Billable Service Quantity** is the service quantity priced on the calculation line. This quantity will be different from the measured product/charge type is there are SQ rules or register rules in effect.
- **Base Amount** is used by calculation lines (e.g. taxes) that are cross-referenced to other calculation lines and whose value(s), therefore, depend on the amounts calculated by those other lines. The Base Amount shows the total amount derived from the cross-referenced line(s) that the current line then used to calculate its billed amount.
- **Sequence** refers to the sequence number of the rate component on the applicable rate version that was used to calculate the line.
- **Meas Peak Qty** is checked if the UOM priced on the calculation line is used to measure a peak quantity.
- **Exempt Amount** is the amount of the calculated charge that the customer doesn't have to pay because they are tax exempt. This will only be calculated if you have provided an contract ID on the main page.

Copyright © 2024, Oracle and/or its affiliates.

- **Distribution Code** is the distribution code associated with the calculation line. This distribution code is used to build the general ledger details on the bill segment's financial transaction.

Rate Check Results - SQ Details

After you have pressed the **Check Rate** button on the **Main** page, you can navigate to the **SQ Details** tab to view the service quantities that supplied to (or derived by) you rate. Please be aware that a rate can derive service quantities if there are register rules, service quantity rules or Variance Parameter mapping rate components.

Description of Page

The **Rate Schedule** and its description, the **Revenue Class** and its description and the **Bill Period** are displayed at the top.

The SQ details grid is a snapshot of the service quantities amassed from:

- The quantities input on the main page OR the quantities taken from the service quantity collection for the bill segment, if it was input on the main page.
- The SQ rules defined on the input rate.
- The register rules defined on the input rate.

The following information is displayed in the grid:

- **UOM** is the unit of measure of the service quantity.
- **Variance Parameter** is the Variance Parameter code of the service quantity.
- **SQI** is the service quantity identifier of the service quantity.
- **Initial Service Quantity** is the initial quantity amassed by the system before application of the rate's SQ rules.
- **Billable Service Quantity** is the service quantity that will be priced by the contract's rate. This amount differs from the initial service quantity when the rate's SQ rules manipulate product/charge type.

Rate Check Results - Characteristics

After you have pressed the **Check Rate** button on the **Main** page, you can navigate to the **Characteristics** tab to view the final characteristics collection as manipulated by your rate's plug-ins.

Description of Page

The **Rate Schedule** and its description, the **Revenue Class** and its description and the **Bill Period** are displayed at the top.

The collection of **Characteristic Types** and **Characteristic Values** reflects any changes made to the initial characteristics collection (input on the main tab page) after applying the rate.

Rate Schedule Summary

The **Rate Schedule Summary** screen allows you to search for rate schedules using various search criteria. It also allows you to view rate components defined for each rate schedule and rate version combination. This screen consists of the following zones:

- Search on page 767
- Rate Components on page 768

Search

The **Search** zone allows you to search for rate schedules using various search criteria. This zone contains the following two sections:

Copyright © 2024, Oracle and/or its affiliates.

- **Search Criteria** - The **Search Criteria** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Rate Schedule	Used to specify the rate schedule.	No
Description	Used to specify the description of the rate schedule.	No

- **Search Results** - On clicking the **Search** button, the search results appear based on the specified search criteria. The **Search Results** section contains the following columns:

Column Name	Column Description
Rate Schedule	Displays the rate schedule.
	Note: It has a link. On clicking the link, the **Rate Schedule** screen appears with the details of the respective rate schedule.
Description	Displays the description of the rate schedule.
Service Type	Indicates the service type associated with the rate schedule.
Frequency	Indicates the frequency in which the charges are expressed for the rate schedule.
Rate Version	Indicates the rate version of the rate schedule.

Related Topics

For more information on...	See...
How to search for a rate schedule	Searching for a Rate Schedule on page 769

Rate Components

The **Rate Components** zone lists the rate components defined for the rate schedule and rate version combination. This zone contains the following columns:

Column Name	Column Description
Rate Schedule	Indicates the rate schedule to which the rate component belongs.
	Note: It has a link. On clicking the link, the **Rate Schedule** screen appears where you can view the details of the respective rate schedule.
Effective Date	Displays the date from when the rate component is effective.
Sequence	Indicates the sequence of the rate component in the rate schedule.
Rate Component Type	Indicates the type of rate component.
Description on Bill	Displays the description that should appear on the customer's printed bill. This description appears when the rate component contributes a line to the customer's bill.

By default, the **Rate Components** zone does not appear in the **Rate Schedule Summary** screen. It appears when you click the **Broadcast** (📶) icon corresponding to the rate schedule in the **Search Results** section.

Related Topics

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
How to view rate components of a rate schedule	Viewing Rate Components of a Rate Schedule on page 769

Searching for a Rate Schedule

Procedure

To search for a rate schedule:

1. Click the **Menu** link in the **Application** toolbar.
   A list appears.
2. From the **Main** menu, select **Rates** and then click **Rate Schedule Summary**.
   The **Rate Schedule Summary** screen appears.
3. Enter the search criteria in the **Search** zone.

   Note: ORMB search engine supports wildcard search, where you can substitute the percentage (%) symbol as a stand in for any word or letter in a search criteria. You can use the '%' wildcard character in all input fields except the date and ID fields. The '%' wildcard character is suffixed automatically at the end of the partial search criteria. Therefore, you may or may not specify the wildcard character at the end of the partial search criteria. However, you have to prefix the wildcard character manually wherever required.
4. Click **Search**.
   A list of rate schedules that meet the search criteria appears in the **Search Results** section.
5. If required, you can export the list of rate schedules in the Excel format by clicking the **Export To Excel** menu option from the **Explorer Zone** (⊡) menu.

Related Topics

For more information on...	See...
Rate Schedule Summary screen	Rate Schedule Summary on page 767
Search zone	Search on page 767

Viewing Rate Components of a Rate Schedule

Procedure

To view the rate components of a rate schedule:

1. Search for the rate schedule in the **Rate Schedule Summary** screen.
2. In the **Search Results** section, click the **Broadcast** (📡) icon corresponding to the rate schedule whose details you want to view.
   The **Rate Components** zone appears.
3. View all rate components defined for the rate schedule and rate version combination in the **Rate Components** zone.
4. If required, you can export the list of rate components in the Excel format by clicking the **Export To Excel** menu option from the **Explorer Zone** (⊡) menu.

Related Topics

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
How to search for a rate schedule	Searching for a Rate Schedule on page 769
Rate Components zone	Rate Components on page 768

Effective Dates & Price Proration

In this section, we describe how proration applies to different types of rate components. Proration implies assessing charges proportionately.

The term "proration" describes two different issues:

- Prorating a charge whose value changes during a bill period. For example, if a tax rate changes during a bill period and you've indicated that such changes should be prorated, the system prorates the tax change (e.g., 20 days at 5% and 10 days at 6%).
- Prorating charges when the time period being billed is not in sync with the time period in which the charges were defined in the rate. For example, if a rate contains a flat monthly charge and the bill period spans two months, the flat charge must be prorated.

This is a complicated topic as it's possible for many proration issues to exist on a single bill. For example, on a single bill:

- The rate structure can change several times during the bill period (i.e., multiple rate versions are effective).
- The taxing authority changes the tax rate.
- The customer's tax exemption changes.
- The customer-specific price can change.
- etc.

How To Add A New Service Quantity Rule Algorithm

CAUTION: Technical Information Below! This section is meant for technicians only!

If the SQ rule algorithms that are supplied with the base package are not sufficient for your rating requirements, you will have to write a new SQ rule algorithm. This section describes how to do this. After you have followed the steps outlined below, you can then set up the control tables necessary to invoke this algorithm as described under Setting Up Service Quantity Rules.

Follow the following steps to define a new service quantity (SQ) rule:

- Assign a two-digit identifier to the SQ rule. This identifier should begin with an `X` or `Y`. For example, you could give your first SQ rule the identifier `X1`, the second `X2`, etc.
- Create an algorithm type for your new SQ rule. This algorithm type exists to define the *types* of parameters supplied to your SQ rule. This algorithm type must be identified by the two-digit identifier allocated above.
- While SQ rules require an *algorithm type*, they do NOT require an *algorithm*. Why? Because you set up an SQ rule to define the values of the parameters passed to your program (and therefore you don't need an algorithm as algorithms exist to define the values of the parameters passed to a program).
- Go to *Look Up Options* and add a new lookup value for the field `SQ_RULE_FLG`. This should have a Field Value equal to the two-digit identifier allocated above.
- Write your new program. This program should be named `CMPRQ`'nn' `X`- where 'nn' is the same two-digit identifier. When you create the new program, please copy an existing SQ rule program (their program names are all in the format CIPRQ_X). This way you will automatically inherit the copybooks necessary to support the routine.

Copyright © 2024, Oracle and/or its affiliates.

CAUTION: Do NOT alter any of the copybooks contained in the linkage section of the COBOL algorithm routine. If you do need extra information for your algorithm you should execute SQL to get the information. If you do alter a linkage section copybook the upgrade process will be greatly complicated. Note, however, that virtually all information available to the rate application is being passed to the rule algorithms - so it is likely the information you need will be at hand.

How To Add A New Register Rule Algorithm

CAUTION: Technical Information Below! This section is meant for technicians only!

If the register rule algorithms that are supplied with the base package are not sufficient for your rating requirements, you will have to write a new register rule algorithm. This section describes how to do this. After you have followed the steps outlined below, you can then set up the control tables necessary to invoke this algorithm as described under Setting Up Register Rules.

Follow the following steps to define a new Register (register) rule:

- Assign a two-digit identifier to the register rule. This identifier should begin with an `X` or `Y`. For example, you could give your first register rule the identifier `X1`, the second `X2`, etc.
- Create an algorithm type for your new register rule. This algorithm type exists to define the types of parameters supplied to your register rule. Refer to Base Package Register Rules for examples. This algorithm type must be identified by the two-digit identifier allocated above.
- While register rules require an *algorithm type*, they do NOT require an *algorithm*. Why? Because you set up a register rule to define the values of the parameters passed to your program (and therefore you don't need an algorithm as algorithms exist to define the values of the parameters passed to a program).
- Go to *Look Up Options* and add a new lookup value for the field `REG_RULE_ALG_FLG`. This should have a Field Value equal to the two-digit identifier allocated above.
- Write your new program. This program should be named `CMPRR`'nn' `X`- where 'nn' is the same two-digit identifier. When you create the new program, please copy an existing register rule program (their program names are all in the format CIPRR_X. This way you will automatically inherit the copybooks necessary to support the routine.

CAUTION: Do NOT alter any of the copybooks contained in the linkage section of the COBOL algorithm routine. If you do need extra information for your algorithm you should execute SQL to get the information. If you do alter a linkage section copybook, the upgrade process will be greatly complicated. Note, however, that virtually all information available to the rate application is being passed to the rule algorithms - so it is likely the information you need will be at hand.

Copyright © 2024, Oracle and/or its affiliates.

Chapter 10

Quotes

Topics:

- The Big Picture Of Quotations
- Maintaining Quotes

A quote allows you to show a customer how their bill segments will look if a given set of pricing terms is applied to a given amount of product/charge type. If the customer accepts a quote, one or more real contracts are created.

Fastpath: Refer to Defining Quotation Options for how to setup the system to enable quotation functionality.

Note: Separate module. Please note that the quoting functionality is part of the `Contract Management` module. If this module is not applicable to your business you may turn it off. Refer to Turn Off A Function Module for more information..

The topics in this section describe the quotation functionality.

The Big Picture Of Quotations

The topics in this section provide background information about a variety of quotation features and functions.

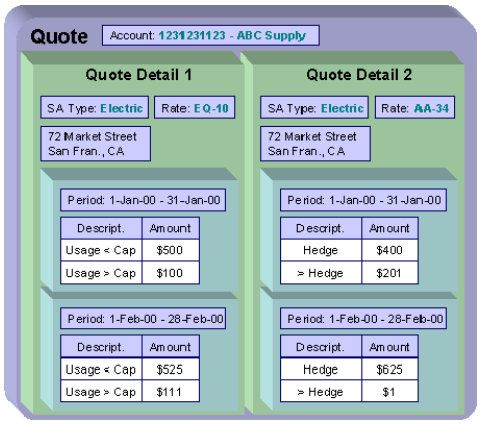
> **CAUTION:** If you are not familiar with the concepts described under Sales and Marketing, Billing, and Rates, the quotation functionality will be difficult to understand.

Basic Quotation Concepts

The section describes basic quotation concepts.

A Quote Shows How Bill Segments Might Look

A quote shows how a customer's bill segments will look if a given set of pricing terms is applied to a given amount of product/charge type. The following example illustrates how a quote might look that provides a customer with two different options for their banking service. If the customer elects either option, a new contract would be created.



Please note the following about a quote:

- A quote is related to an account.
- A quote can contain many quote details. A quote detail shows how bill segments will look if a given set of pricing terms (e.g., rates, riders, contract quantities, time-of-use-maps, contract options, etc.) is applied to a given amount of consumption.
- A quote detail may contain many simulated bill segments. For example, a quote detail could show the customer what their monthly bills would be if a given set of pricing terms were used over the course of several months (where each month has different consumption levels and different prices).
- A quote may contain multiple quote details for the same service. For example, a quote can contain four different pricing options for the same service (where each option is shown using a separate quote detail). The customer can then compare them and choose the optimum option.

Proposal Contracts Must Exist

You cannot create a quote unless its account has at least one *proposal* contract. A proposal contract contains the pricing terms, legal terms, and template consumption used to generate the simulated bill segments that appear on a quote. It might be helpful to view a proposal contract as special type of contract that is ignored when a bill is produced for an account.

The topics in this section provide detailed information about proposal contracts.

Copyright © 2024, Oracle and/or its affiliates.

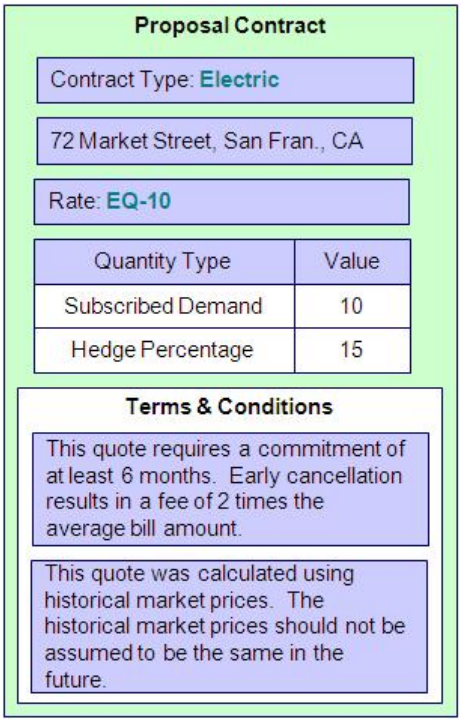
Proposal Contracts Contain Pricing Terms

The pricing terms (e.g., rates, contract quantities, riders, bill factor values, load curves, etc.) used to generate a quote detail's simulated bill segments reside on a special type of contract (contract) referred to as a "proposal contract". Proposal contracts are like real contracts in many respects. In fact, you can view and change a proposal contracts pricing terms using the same transaction used to maintain real contracts (i.e., contract maintenance).

> **Note: Quote details are rather empty.** A quote detail doesn't contain the pricing terms or template product/charge type. Rather, a quote detail references a proposal contract and the proposal contract contains the pricing terms and template product/charge type.

Legal Terms and Conditions May Be Specified On Both Proposal And Real Contracts

In addition to containing pricing terms, proposal contracts also contain the legal terms and conditions (T&Cs) that govern the proposal.



Your master list of terms and conditions are defined when you setup your control tables.

> **Note: Real contracts and proposal contracts can reference T&Cs.** It should be noted that both proposal and real contracts can reference T&Cs. The T&Cs on a proposal contract are those that will apply if the customer accepts the related quote detail. The T&Cs on a real contract are those that apply while the contract is active.

Proposal Contracts Contain Billing Scenarios And Template Consumption

In addition to the pricing terms and the legal terms and conditions, a proposal contract also contains billing scenarios. A billing scenario defines:

- The number of simulated bill segments that will be generated when a quote detail is produced for the proposal contract.
- The bill period of each simulated bill segment.
- The amount of product/charge type consumption to be priced under each simulated bill segment. Note this statement is only true for non-interval product/charge type consumption.

A proposal contract's billing scenarios can be created as follows:

Copyright © 2024, Oracle and/or its affiliates.

- A plug-in spot exists on *contract type* called `Proposal Contract Creation`. Algorithms plugged into this spot can setup billing scenarios. Refer to *Enabling The Automatic Generation Of Billing Scenarios* for more information about how to do this.
- A user can add / change / delete billing scenarios on a proposal contract using Contract - Billing Scenario.
- You could design a workflow process to initiate a request to the customer to send their template product/charge type consumption to you. When the customer sends this information to you (via a notification), the proposal contract's billing scenarios can be setup. Refer to Workflow and Notifications for more information.

Proposal Contracts Differ From Real Contracts

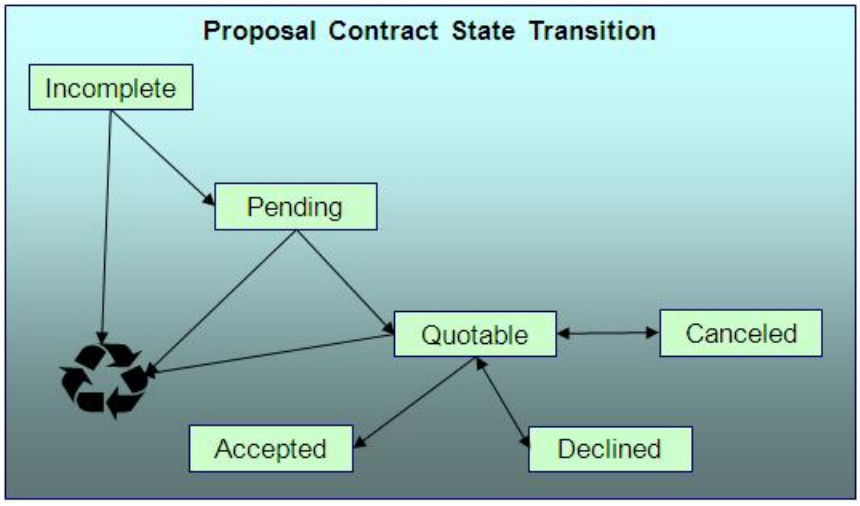
A proposal contract is just like any other contract in that it contains the pricing terms that are used to generate how much the customer owes for service. In fact, the contract maintenance transaction is used to maintain both types of contracts (however, you cannot use this transaction to create a proposal contract; proposal contracts can only be created using the order transaction).

However, there are differences between proposal contracts and real contracts:

- Billing ignores proposal contracts.
- Proposal contracts cannot have financial transactions.
- Proposal contracts are not subject to credit and collections.
- Proposal contracts do not appear on lists of "active" contracts in the system's various zones, grids and trees.
- A proposal contract's state transition is very different from a real contract. Refer to The Lifecycle Of A Contract for an illustration of a real contracts state transition. Refer to Proposal Contract State Transition for an illustration of a proposal contracts state transition.

Proposal Contract State Transition

The following diagram shows the possible lifecycle of a proposal contract.



You should never see an `Incomplete` proposal contract as this is a transitory state used when a proposal is first initiated using the Order transaction. The system always transitions such proposal contracts to the `Pending` state. Please note that when a proposal contract becomes `Pending`, the system executes any `Proposal Contract Creation` algorithms plugged in on the proposal contract's contract type. You might use such algorithms to generate billing scenarios or initiate a workflow process that requests template consumption from the prospect.

A proposal contract exists in the `Pending` state while you are working on the proposal. For example, you might want to leave the proposal contract in this state while you wait for the prospect to send you their historical consumption.

A proposal contract is `Quotable` when a quote detail can be generated for it (at least one billing scenario must be linked to the proposal contract before it can become `Quotable`). Note, proposal contracts can be modified when they are `Quotable`.

Copyright © 2024, Oracle and/or its affiliates.

Note: Automatically transitioning a proposal contract to the Quotable state when it's first created. As described earlier, you can setup a `proposal contract creation` algorithm to automatically create billing scenarios when a proposal contract is created. The base package algorithm can also automatically transition the proposal contract to the `Quotable` state (thus making it ready for quotation). Refer to Enabling The Automatic Generation Of Billing Scenarios for more information.

A proposal contract becomes `Accepted` when a customer accepts one of its quote details.

A proposal contract becomes `Declined` when a customer declines one of its quote details (or its quote expires). A `Declined` proposal contract can be "reopened" (by making it `Quotable`). You may need to do this if you want to change a proposal contract's terms in preparation of generating a new quote detail.

A proposal contract becomes `Canceled` when a quote detail is withdrawn (or its quote expires). You may need to do this if you want to change a proposal contract's terms in preparation of generating a new quote detail.

`Incomplete`, `Pending` and `Quotable` proposal contracts may be deleted.

Orders Are Used To Create Proposal Contracts

Proposal contracts can only be created using the order transaction. This transaction allows a user to pose eligibility-oriented questions to a customer or prospect. The system determines which packages of goods and services for which they are eligible based on the answers to the questions. If the customer is interested in receiving a package, one or more contracts (real and/or proposal) are created using the Contracts To Create information defined on the selected package.

Note: You must setup campaigns and packages before you can create an order. If you are not familiar with the order transaction, you should take the time to form an intuitive understanding of the concepts described under Sales & Marketing. With this knowledge in place, you can design and set up the campaigns and packages necessary to allow an order to create proposal contracts. Keep in mind that a single switch controls whether real contracts or a proposal contracts are created when a package is selected on the order transaction. This switch resides on Package - Contracts To Create.

Note: A package could create a combination of real and proposal contracts. When a customer elects to take a package of goods and services (via the order transaction), the system creates a contract for each contract type defined on the package's Contracts To Create. It is possible to setup a package to create both real and proposal contracts.

Note: You don't create special contract types for proposal contracts. The contract type on a proposal contract is the same as the contract type on the real contract that's created when the proposal contract is accepted. This is because the business rules on your contract types govern both real and proposal contracts.

Creating Quotes And Quote Details

Once an account has `quotable` proposal contracts, you can generate a quote for the account using the quote maintenance transaction. When you generate a quote for an account, you can include quote details for any / all of the account's `quotable` proposal contracts.

The logic used to create a quote detail's simulated bill segments is not hard-coded. Rather, the system calls the algorithm plugged in on the proposal contract's contract type in the `Proposal Contract - Bill Segment Generation` plug-in spot. Refer to *Enabling The Generation Of Simulated Bill Segments* for information about this plug-in spot.

The major difference between a bill segment linked to a quote detail and one linked to a bill is that a quote detail's bill segments are calculated using simulated product/charge type consumption. This means that the financial information on quote details is not a guess - it's what the bill would be if a given amount of product/charge type consumption was priced using the pricing terms (as long as you opt to use rate application to calculate the simulated bill segments).

In addition to this major difference, there are a few minor differences between a real bill segment and a simulated bill segment:

- A quote detail bill segment does not have a foreign key to a bill. Rather, it references a quote detail.
- A quote detail bill segment's prime key is inherited from its quote detail (rather than its bill).
- A quote detail bill segment can never have a financial transaction linked to it.

Copyright © 2024, Oracle and/or its affiliates.

- A quote detail bill segment's state transition is simpler (it's either in the `Error` state or `OK`).

Completing A Quote Creates Quote Routings (And More)

When a quote's quote details are all `OK`, you change the status of the quote to `complete` to indicate the quote is ready to be sent to the customer or prospect.

When a quote is completed, the system creates a quote routing for each person who receives a copy of the account's quotes. When you indicate a person should receive a copy of the account's quotes, you define the routing method (e.g., email, fax, post) and the appropriate address.

Note: Recipients of quotes. You define who receives a copy of an account's quotes on the Account - Persons page. The system assumes (by default) that the main person linked to the account should receive the account's quotes. You can subsequently indicate any person linked to the account should receive its quotes (and turn off the creation of quotes for the main person). In addition, you can manually add additional quote routings to a quote if you need to send the quote to individuals not currently linked to the account. Refer to How To Reprint A Quote for more information.

The information that appears on the quote routing record is controlled by the *quote route type* specified on the respective account / person information:

- If the quote route type indicates the person's quotes are routed via *fax*, the person's fax number is used as the routing information. The system knows which of a person's phone numbers is a fax number by the phone type.
- If the quote route type indicates the person's quotes are routed via *email*, the person's *Email address* is used as the routing information.
- If the quote route type indicates the person's quotes are routed via the *postal* service, the address constituents are populated with the address specified on the ***bill*** address sourcethat is also specified on the account / person information. Note, if the person has a seasonal address effective on the business date, the seasonal address will be used regardless of the value of **bill address source**.

Regardless of whether the quote is routed via fax, email or post; the routing record is also stamped with the batch process and run number that performs the download of the quote information. The batch process is defined on the quote route type. Refer to Printing Quotes for more information about this batch process.

In addition to creating routing information at completion time, the system executes any quote completion algorithms plugged in on the account's customer class. For example, you may want to create a workflow process to manage the processing of the quote. To do this, you plug in a Quote Completion Algorithm that creates a workflow process on the respective *customer class*.

Accepting / Declining Quote Details

A customer can accept or decline each quote detail on a quote.

When a quote detail is accepted, the system:

- Changes the state of the related proposal contract to `accepted`.
- Executes the `Proposal Contract - Acceptance` algorithms plugged in on the proposal contract's contract type. The base package algorithm creates a real contract by copying the information from the proposal contract. The real contract is created in the `pending start` state. This new contract behaves similarly to a `pending start` contract that's created via the Order or Start/Stop transactions.

When a quote detail is declined, the user must supply a decline reason. The system then changes the state of the related proposal contract to `declined`.

Fastpath: Refer to Proposal Contract State Transition for more information about a proposal contract's states.

Note: Reopening a declined proposal contract. If a customer changes their mind about a `declined` proposal contract, you can return it to the `quotable` state using the contract maintenance transaction.

Note: Rejecting an entire quote. If all details on a quote are being declined, you can simply change the state of the quote to `closed` rather than reject each quote detail. Refer to Closing Quotes for more information.

Copyright © 2024, Oracle and/or its affiliates.

Tinkering With Proposal Contracts After Quote Generation

The system allows a `quotable` proposal contract to be changed, EVEN IF QUOTE DETAILS HAVE BEEN GENERATED. This is intentional as otherwise you would have to create an entirely new proposal contract if you wanted to send another quote for slightly different terms. Your technical team could always introduce a user exit to prevent this from happening if you really didn't want this to take place.

Because the system allows this, a potential issue is that the customer may end up accepting a proposal contract that differs from the terms on his quote (e.g., you could change the contract after the quote detail is generated). For this reason, we snapshot the version number of the proposal contract on the quote detail at the time it's generated. If a proposal contract is accepted and its version number differs from the value on its quote details, a warning is produced.

A warning is also issued if you change a proposal contract and a quote detail exists (reminding you to generate a new quote / quote detail).

In addition to the above, it's also important to understand that the system intentionally allows multiple quotes to reference a given proposal contract. This allows you to leave multiple quotes open that reference the same proposal contract(s).

Closing Quotes

Every quote has an expiration date. The CLOSEQTE background process `closes` quotes on their expiration date. This process changes the state of all related proposal contracts to `canceled` or `rejected` (an input parameter of this process controls which state is used).

You can also close a quote real time using the quote maintenance transaction. When you do this, the system asks if the related proposal contracts should be left as `quotable` or if they should be changed to `canceled` or `rejected`.

> **Note: Reopening a rejected proposal contract.** If a customer changes their mind about a `rejected` proposal contract, you can return it to the `quotable` state using the contract maintenance transaction.

Alerts

The alert zone in the dashboard indicates whether:

- The selected account has `pending` or `complete` quotes. Refer to Quote Lifecycle for more information.
- The selected account has `pending` or `quotable` proposal contracts. Refer to Proposal Contract State Transition for more information.

Printing Quotes

The contents of this section describe the technical implementation of printing quotes.

Quote Routings Are Created For Each Recipient Of A Quote

Refer to Completing A Quote Creates Quote Routings for a description of how the system constructs the information used to route a quote to one or more recipients.

Quote Route Types Control The Information Merged Onto Quotes

Every quote routing references a *quote route type*. The route type controls the following quote functions:

- It contains an algorithm that is responsible for extracting the information merged onto your quotes. Specifically, algorithms of this type create the "flat file records" that are passed to your quote-print software. Algorithms of this type are called under the following scenarios:
  - The background process that builds the flat file that's passed to your quote-print software calls these algorithms to construct the "flat file records" for each quote. Refer to Technical Implementation of Printing Quotes In Batch for the details.
  - If your quote-print software has the ability to construct a real-time image of a quote (in a PDF), you can plug in an `Online Quote Image` algorithm on the Installation Record. This type of algorithm calls the quote's

Copyright © 2024, Oracle and/or its affiliates.

dispatch group extract algorithm to extract the information that is merged onto the quote. Refer to Technical Implementation Of Online Quote Image for the details.

- Refer to The Doc 1 Records for examples of the record layouts passed to the Doc 1 print software.
- It contains the ID of the background process that builds the flat file that's passed to your quote-print software. The base package example of this process (known by the batch control ID of QUOTROUT) simply calls the extract algorithm on the routing record's route type to format the information placed onto the flat file.

Technical Implementation Of Online Quote Image

Users can view an image of any quote if you set up the following:

- Plug in an `Online Quote Image` construction algorithm on the Installation Record. Refer to ONQD-DOC1 for an example of such an algorithm. Note if your quote-print software is not capable of producing a PDF containing an image of a quote, users will not be able to view images of quotes.
- Plug in the appropriate extract algorithm on each *quote route type*. Algorithms of this type format the records that contain the information that is passed to your printing software. Refer to QTEX-EX for an example of such an algorithm.

When you plug in these algorithms, a button appears on Quote - Main. When a user clicks this button, the following takes place:

- The installation record's `Online Quote Image` construction algorithm is executed.
- This algorithm calls the quote's route type's extract algorithm. This algorithm constructs the information that's merged onto the quote and returns it to the `Online Quote Image` algorithm. This algorithm, in turn, passes it to the quote-print software.
- The quote-print software renders an image of the quote in a PDF and returns it to the `Online Quote Image` algorithm.
- And finally, the `Online Quote Image` algorithm displays the PDF in a browser session. Note that the client must have Adobe Acrobat Reader installed to view PDF files.

Technical Implementation Of Printing Quotes In Batch

The batch process that extracts quote information reads all quote routing records in a given run number that are marked with its batch control ID. For each quote routing record, it creates numerous records on a flat file. These records contain the information that is merged onto your quotes.

The base package example of this process (known by the batch control ID of QUOTROUT) simply calls the extract algorithm on the routing record's route type to format the information placed onto the flat file. Refer to Quote Route Types Control The Information On Quotes for more information.

Note: If your software doesn't support online quote images. The algorithm that formats quote extract records that's plugged in on *quote route type* serves two purposes: 1) it formats the records used to construct online images of a quote, and 2) it formats the records downloaded to your quote print software in batch. If your quote printing software does not support the rendering of quote images real time, there is no need to create an extract algorithm. Rather, you should simply develop your own download process that both formats the extract records and downloads them (and then specify this batch process on your quote route type).

Reproducing The Quote Print Flat File

You can reproduce the flat file at any time. Simply request the quote production background process and specify the run number associated with the historic run.

How To Reprint A Quote (For The Original Recipients or For Someone New)

A copy of a quote is sent to every individual / business specified in the quote's routing details. The system creates the routing details when a quote is completed. Refer to Completing A Quote Creates Quote Routings for more information about how this information is compiled during quote completion.

If you want to reprint a quote OR if you want to send a quote to someone other than the original recipients:

Copyright © 2024, Oracle and/or its affiliates.

- Navigate to the Quote - Main page.
- Transfer to the Quote - Quote Routings tab.
- Insert a new row (click the + button). Note that you can only insert information on a quote that is `Complete`.
- Specify the name and address to which the quote should be sent. If you want to send the quote to someone other than the original recipient, specify the individual's name and address.
- Save the quote.

If a quote is `Complete` but the print process hasn't yet executed, you may change the name and address on the original routing details; you don't have to insert a new routing row.

The Doc 1 Records

Numerous different types of records are interfaced to the Doc 1 quote print software. However, they all share a common structure:

- The first 4 bytes are called the **Printer Key**. This field is a record type used by the Doc 1 quote template.
- The next X bytes are called the **Sort Key**. This field is used to ensure the various records used to construct a printed quote appear in the correct order in the interface file.
- The next 12 bytes are called the **Mailing Key**. This field is used to control the order in which the formatted quotes are printed.
- The remaining bytes contain the **Quote Information** that appears on the printed quote. The type of information differs for each type of record.

The topics in this section describe each component.

Printer Key (Record Types)

The following table describes the different types of records in the quote print flat file:

Record Type	Description	Notes
0010	Global extract information	Required. 1 per flat file
0100	Quote record	Required. 1 per Quote.
0200	Quote Detail record	1 per quote detail.
0300	Terms and Conditions record	1 per terms and conditions linked to the proposal contract.
0400	Bill segment record	1 per bill segment linked to the quote detail.
0500	Bill calculation header	1 per bill calculation header linked to the bill segment.
0600	Bill calculation line	1 per bill calculation line linked to the bill calculation header.
0700	End of record type 0500	1 for each bill calculation header
0900	SQ detail	1 for each service quantity detail linked to the bill segment.
1000	End of record type 0400	1 for each bill segment.
1100	End of record type 0200	1 for each quote detail.
9999	End of record type 0100	1 for each quote.

Sort Key

The following table describes the structure of the sort key that appears on the print record of each quote:

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
Quote ID	A12	
Sequence Number	N2	The sequence number of the account / person information associated with this quote routing information.
Quote record group	A2	10 Quote 20 Quote Detail 30 Quote end
Quote Detail ID	A12	
Quote Detail record group	A2	10 Quote Detail record 20 Contract Terms and Conditions 30 Bill Segment 40 Quote Detail record end
Contract T&C Short Description	A60	
Bill Segment End Date	A10	
Bill Segment Id	A12	
Bill Segment Group	A2	10 Bill segment 20 Bill segment calculation group 30 Bill segment read detail 40 Bill segment SQ detail 50 Bill segment end
Bill Calc Header Sequence	N2	
Bill Calc Header Group	A2	10 Bill calculation header 20 Bill calculation line 30 Bill calculation end
Line Seq	N4	Defines order of rows for similar record types

Mailing Key

The following table describes the structure of the mailing key that appears on each quote print record:

Field Name	Format	Source/Value/Description
Postal code	A12	This is the postal code on the quote routing record.

Quote Information

The topics in this section describe the information that appears on each on each bill print record.

Global Extract Information (0010) Record Definition

This is a "header" record that contains information about the extract run as a whole.

Copyright © 2024, Oracle and/or its affiliates.

Note: Please be aware that if you do not sort your extract file by **CIS Sort Key** after it is produced, this record will physically be the last record on the extract file.

The following table lists the header elements that contain information about the extract run:

Field Name	Format	Source/Value/Description
BATCH_CD	A8	Batch Control Code of the extract process.
BATCH_NBR	N10	Batch Number of the extract process
BATCH_RERUN_NBR	N10	Batch Rerun number of the extract process.
QTE_CNT	N10	Count of quotes on this extract.
EXTRACT_DTTM	A26	System time of extraction.
QUOTE_ID_START	A12	This is only used when a quote is produced as a result of an online request (from Quote - Main). It contains the Quote Id to extract.
QUOTE_ID_END	A12	This is only used when a quote is produced as a result of an online request (from Quote - Main). It contains the Quote Id to extract.
THREAD_NBR	N10	This field is only used if quotes are extracted in batch. It contains the thread number in which the quote was extracted.
THREAD_CNT	N10	This field is only used if quotes are extracted in batch. It contains the total number of threads that were extracted.

Quote (0100) Record Definition

The following table describes the fields that appear in the quote record:

Field Name	Format	Source/Value/Description
QUOTE_ID	A12	CI_QUOTE
ACCT_ID	A10	CI_QUOTE
EXPIRE_DTTM	A26	CI_QUOTE
QUOTE_STAT_FLG	A2	CI_QUOTE
CURRENCY_CD	A3	CI_ACCT
CUR_SYMBOL	A4	CI_CURRENCY_CD
ENTITY_NAME1	A64	CI_QUOTE_RTG. Mailing Entity Name1.
ENTITY_NAME2	A64	CI_QUOTE_RTG. Mailing Entity Name2.
ENTITY_NAME3	A64	CI_QUOTE_RTG. Mailing Entity Name3.
ADDRESS_SBR	A361	CI_QUOTE_RTG. Refer to its components in a later section.

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
EMAILID	A70	CI_QUOTE_RTG
PHONE-FAX	A64	CI_QUOTE_RTG

Quote Detail (0200) Record Definition

The following table describes the fields that appear in the quote detail record:

Field Name	Format	Source/Value/Description
QUOTE_DTL_ID	A12	CI_QUOTE_DTL
SA_ID	A10	CI_QUOTE_DTL
TOTAL_AMT	N13.2	CI_BSEG_CALC. Sum of all calculation amounts (quote detail's bill segments).
ADDRESS_SBR	A361	CI_PREM. Refer to its components in a later section.
SA_TYPE_DESCR	A60	CI_SA_TYPE_L
SVC_TYPE_CD	A2	CI_SVC_TYPE_L
SA_VERSION	N5	CI_QUOTE_DTL
QTE_DTL_SA_VERSION	N5	CI_QUOTE_DTL
RS_DESCR	A60	CI_RS_L. Description of Rate effective on contract start date.

Contract Terms and Conditions (0300) Record Definition

The following table describes the fields that appear in the contract's terms and conditions record:

Field Name	Format	Source/Value/Description
START_DT	A10	CI_SA_TC
END_DT	A10	CI_SA_TC
TC_TYPE_CD	A12	CI_TC
DESCR	A60	CI_TC_L
TC_TYPE_TXT	A2000	CI_TC_L

Bill Segment (0400) Record Definition

The following table describes the fields that appear in the bill segment record:

Field Name	Format	Source/Value/Description
START_DT	A10	CI_BSEG
END_DT	A10	CI_BSEG
BSEG_STAT_FLG	A2	CI_BSEG
BSEG_AMT	S13.2	CI_BSEG_CALC_LN

Bill Calculation Header (0500) Record Definition

The following table describes the fields that appear in the bill calculation header record:

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
START_DT	A10	CI_BSEG_CALC
END_DT	A10	CI_BSEG_CALC
RS_CD	A8	CI_BSEG_CALC
CALC_AMT	S13.2	CI_BSEG_CALC
DESCR_ON_BILL	A80	CI_BSEG_CALC

Bill Calculation Line (0600) Record Definition

The following table describes the fields that appear in the bill calculation line record:

Field Name	Format	Source/Value/Description
APP_IN_SUMM_SW	A1	CI_BSEG_CALC_LN
CALC_AMT	S13.2	CI_BSEG_CALC_LN
EXEMPT_AMT	S13.2	CI_BSEG_CALC_LN
UOM_CD	A4	CI_BSEG_CALC_LN
TOU_CD	A8	CI_BSEG_CALC_LN
SQI_CD	A8	CI_BSEG_CALC_LN
BILL_SQ	S12.6	CI_BSEG_CALC_LN
DESCR_ON_BILL	A80	CI_BSEG_CALC_LN

End of Bill Calculation Header (0700) Record Definition

The following table describes the fields that appear at the end of the bill calculation header record:

Field Name	Format	Source/Value/Description
Dummy field		

Bill Segment Read (0800) Record Definition

The following table describes the fields that appear in the bill segment read record:

Field Name	Format	Source/Value/Description
SP_ID	A10	CI_BSEG_READ
BADGE_NBR	A16	CI_MTR
UOM_CD	A4	CI_BSEG_READ
TOU_CD	A8	CI_BSEG_READ
SQI_CD	A8	CI_BSEG_READ
RDG_DEC_POS	N1	CI_REG. Decimal positions of the reads.
START_REG_READING	S9.6	CI_BSEG_READ
START_READ_DTTM	A26	CI_BSEG_READ
START_ESTIMATE_SW	A1	CI_REG_READ
END_REG_READING	S9.6	CI_BSEG_READ
END_READ_DTTM	A26	CI_BSEG_READ

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Format	Source/Value/Description
END_ESTIMATE_SW	A1	CI_REG_READ
MSR_QTY	S12.6	CI_BSEG_READ
USAGE_FLG	A2	CI_BSEG_READ
USE_PCT	N3	CI_BSEG_READ
HOW_TO_USE_FLG	A2	CI_BSEG_READ
CONSUM_SUB_FLG	A2	CI_REG
MSR_PEAK_QTY_SW	A1	CI_BSEG_READ
REGISTER_CONST	S6.6	CI_BSEG_READ
QTY_DEC_POS	N1	CI_UOM. Decimal positions of the final quantity.
FINAL_UOM	A4	CI_BSEG_READ
FINAL_TOU	A8	CI_BSEG_READ
FINAL_SQI	A8	CI_BSEG_READ
FINAL_REG_QTY	S12.6	CI_BSEG_READ

Bill Segment SQ (0900) Record Definition

The following table describes the fields that appear in the bill segment SQ record:

Field Name	Format	Source/Value/Description
UOM_CD	A4	CI_BSEG_SQ
TOU_CD	A8	CI_BSEG_SQ
SQI_CD	A8	CI_BSEG_SQ
SQ_DEC_POS	N1	CI_BSEG_SQ
BILL_SQ	S12.6	CI_BSEG_SQ

End of Bill Segment (1000) Record Definition

The following table describes the fields that appear at the end of the bill segment record:

Field Name	Format	Source/Value/Description
Dummy field		

End of Quote Detail (1100) Record Definition

The following table describes the fields that appear at the end of the quote detail record:

Field Name	Format	Source/Value/Description
Dummy field		

End of Quote (9999) Record Definition

The following table describes the fields that appear at the end of the quote record:

Field Name	Format	Source/Value/Description
Dummy field		

Copyright © 2024, Oracle and/or its affiliates.

Address Sub-record Definition

The address appears in both the quote record (0100) and the quote detail record (0200).

The following table describes the fields that appear in the address record:

Field Name	Format	Description
COUNTRY	A3	
ADDRESS1	A64	Refer to Completing A Quote Creates Quote Routings for a description of conditions when this field will contain a fax number or an Email address rather than the first line of a postal address (this only occurs for a bill record (0100)).
ADDRESS2	A64	
ADDRESS3	A64	
ADDRESS4	A64	
CITY	A30	
NUM1	A6	
NUM2	A4	
HOUSE_TYPE	A2	
COUNTY	A30	
STATE	A6	
POSTAL	A12	
GEO_CODE	A11	
IN_CITY_LIMIT	A1	

Maintaining Quotes

A quote allows you to show a customer how their bill segments will look if a given set of pricing terms is applied to a given amount of product/charge type consumption. If the customer accepts the quote, real contracts are created.

Fastpath: Refer to The Big Picture Of Quotations for background information about quotes.

The topics in this section describe how to maintain quotes.

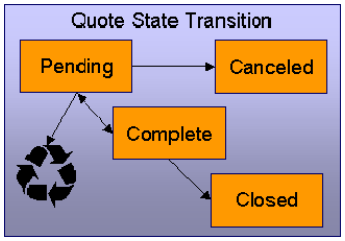
The Lifecycle Of A Quote And Quote Detail

The topics in this section describe the lifecycle of the various quote objects.

Quote Lifecycle

The following diagram shows the possible lifecycle of a quote.

Copyright © 2024, Oracle and/or its affiliates.



`Pending` A quote starts its life in the `Pending` state. Refer to Creating Quotes And Quote Details for more information.

A quote remains `Pending` until you `Cancel` or `Complete` it.

You may delete a `Pending` quote. Deleting a pending quote also deletes its quote details.

`Canceled` You cancel a quote if you want a record of a `Pending` quote that was never sent to a customer.

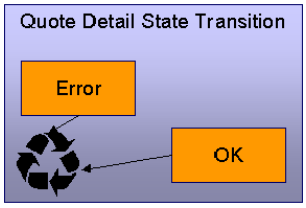
`Complete` A quote becomes `Complete` when it is ready to be sent to a customer. No changes may be made to a `Complete` quote. Refer to Completing A Quote for a description of what happens when a quote is completed.

A `Complete` quote may be reopened. You would only do this if you need to prevent blatant errors from being sent to the customer. For example, if you notice an incorrect quote detail, you could reopen the quote, regenerate or delete the erroneous quote details, and then recomplete the quote.

`Closed` A `Complete` quote becomes `Closed` when its terms are no longer valid. Refer to Closing Quotes for more information.

Quote Detail Lifecycle

The following diagram shows the possible lifecycle of a quote detail.



When a quote detail is generated, it is placed in the `Error` or `Ok` state depending on whether the system successfully generated simulated bill segments for each billing scenario linked to the related proposal contract. A quote detail may be deleted if its parent quote is `Pending`. Refer to Creating Quotes And Quote Details for more information.

Quote - Main

This page is used to maintain quotes. This page provides many diverse functions:

- You can generate, complete, close, reopen, and cancel a quote.
- You can regenerate and delete quote details.
- You can accept, decline and cancel the proposal contracts that contribute a quote detail to the quote.

You should not attempt to use this transaction until you are comfortable with the concepts described under The Big Picture of Quotations.

Open this page using **Sales & Marketing, Quotes**.

Description of Page

Quote Info contains a concatenation of important information about the quote (e.g., its expiration date and status). **Quote ID** is the system-assigned unique identifier of the quote. These values only appear after the quote is added to the database.

Copyright © 2024, Oracle and/or its affiliates.

Account ID identifies the customer or prospect associated with the quote. The name of the account's main customer appears adjacent.

Quote Status contains the state of the quote. Refer to Quote Lifecycle for information about the valid values and the state transition rules.

If your system has been configured to display an online image of a quote, a **Display Quote** button appears adjacent to the quote's status. When clicked, an online image of the quote is displayed. Refer to Technical Implementation Of Online Quote Production for how to enable this feature.

Expiration Date / Time identifies when the quote's terms are no longer valid. Refer to Closing Quotes for how the system uses this information to automatically close a quote.

Refer to Quote Actions for a description of each **Quote Action** button.

This transaction has sophisticated logic that can be used to perform mass updates to the quote's quote details and their related proposal contracts. These mass update actions are executed by selecting quoted details (in the grid) and then clicking the desired action button at the bottom the page. The remainder of this section describes how to use this mass update functionality.

The **Select All**/ **Clear All** buttons are used to select / unselect the quote details. These buttons are disabled when the quote is `Closed`.

Note: 100 quote details at a time. Clicking **Select All** selects the first 100 quote details in the grid. If more than 100 quote details exist, you must select them in batches.

The grid that follows contains the quote details linked to the quote. The following information appears in the grid:

- Select box. Before you can perform any of the mass update actions at the bottom of the page, at least one quote detail must be selected. This field is disabled when the quote is `Closed`. You can use the **Select All** and **Clear All** buttons to change the value of the switch for all quote details in the grid.
- Click the go to button to transfer to the Quote - Quote Detail tab with the corresponding quote detail positioned in the scroll. On this page, you can see the details of the quote detail's simulated bill segments (and, from there, drill down into the details of each bill segment).
- The **Quote Detail Info** column contains a concatenation of a quote detail's division, contract type, status and start date information (this information comes from the quote detail's proposal contract).
- The **Total Amount** column contains the sum of the quote detail's simulated bill segments.
- The **Remarks** column highlights special circumstances associated with the quote detail.
  - `Proposal contract has been changed since the quote was created` appears if the proposal contract was modified after the quote detail was generated. If you'd like to recalculate this quote detail, select this row and click the **Regenerate** button at the bottom of the page to regenerate the quote detail's simulated bill segments.
  - `Quote detail contains bill segments in error` appears if any of the simulated bill segments linked to the quote detail are in `error`. Click the adjacent go to button to transfer to the Quote - Quote Detail tab where the state of each simulated bill segment is shown. You can correct simulated bill segments that are in `error` by **Regenerating** the quote detail (and you do this by selecting the quote detail and clicking the Regenerate button at the bottom of the page).

The **Quote Detail Actions** and **Proposal Contract Actions** can be used to perform mass updates to the selected quote details in the grid. Refer to Quote Detail Actions and Proposal Contract Actions for the details.

Action Buttons On Quote Main

The topics in this section describe the various action buttons on the Quote - Main page.

Quote Actions

The topics in this section describe the various **Quote Action** buttons on the Quote - Main page.

Copyright © 2024, Oracle and/or its affiliates.

Generate

The **Generate** button causes one or more quote details to be created for the account's `Quotable` proposal contracts. This button is enabled when there is at least one `Quotable` proposal contract linked to the selected **Account ID**.

When clicked, the **Generate Quote** window opens.

You must specify the following parameters to generate quote details:

- Indicate if quote details should be created for `All Proposal Contract's` or for a `Specific Proposal Contract`.
- If you indicated a quote detail should be generated for a Specific Proposal Contract, also specify the **Proposal Contract ID**.

After specifying the parameters, click **Generate** to create the quote details for `quotable` proposal contracts.

Fastpath: Refer to Creating Quotes And Quote Details for more information.

CAUTION: If the quote has existing quote details for the designated proposal contracts, they are deleted when you generate quote details (i.e., the system regenerates existing quote details). You can also regenerate quote details by selecting them on the Main page and clicking the **Regenerate** button.

Complete

The **Complete** button causes the quote to become `complete`. This button is enabled when the quote is `pending` and all quote details linked to the quote are `OK`.

Fastpath: Refer to Completing A Quote Creates Quote Routings for a description of what happens during quote completion.

Close

A quote is closed when its terms are no longer valid (i.e., when the related proposal contracts can no longer be accepted). Most quotes are closed automatically by the system (refer to Closing Quotes for how this takes place).

If you want to close a quote real time, click the **Close** button. When this button is clicked, the system opens the **Close Quote** window and asks if anything should be done with the related proposal contracts.

In the resulting window, you must specify one of the following **Proposal Contract Actions**:

- Choose `No Action` if you don't want to change the status of the related proposal contracts.
- Choose `Decline` if you want the status of the proposal contracts that have contributed quote details to the quote to be changed to `declined`. You must also define a **Decline Reason**.
- Choose `Cancel` if you want the status of the proposal contracts that have contributed quote details to the quote to be changed to `canceled`.

Reopen

The **Reopen** button causes a quote to return to the `pending` state. You would only reopen a quote if you need to prevent blatant errors from being sent to the customer. For example, if you notice an incorrect quote detail, you could reopen the quote, regenerate or delete the erroneous quote detail, and then recomplete the quote.

This button is enabled if the quote is `complete`.

Cancel (Quote)

The **Cancel** button causes the quote to become canceled. You would cancel a quote if you want a record of a `pending` quote that was never sent to a customer. This button is enabled when the quote is `pending`.

Canceling the quote does NOT cancel the related proposal contracts. If you want to cancel the related proposal contracts, you have the following options:

- You can click the Close button rather than the **Cancel** button. When you close a quote, you are given the option of also canceling the proposal contracts that contributed a quote detail to the quote.

Copyright © 2024, Oracle and/or its affiliates.

- Before canceling the quote, you can select the quote details and click the Cancel button at the bottom of the page (in the Proposal Contract Actions section). When you click this button, the system cancels the proposal contracts linked to the quote details.
- You can transfer to the Quote - Quote Details tab and then drill down to the each proposal contract (using the contract context menu button). This transfers you to Contract - Main where you can cancel the proposal contract.

Quote Detail Actions

The topics in this section describe the various **Quote Detail Action** buttons on the Quote - Main page. These buttons are used to perform a given action to the quote details that are selected in the quote details grid.

Regenerate

The **Regenerate** button is used to delete and recreate the selected quote details. This button is enabled if the quote is `pending` and at least one row in the quote details grid is selected.

Delete

The **Delete** button is used to delete the selected quote details. This button is enabled if the quote is `pending` and at least one row in the quote details grid is selected.

Proposal Contract Actions

The topics in this section describe the various **Proposal Contract Action** buttons on the Quote - Main page. These buttons are used to perform a given action to the proposal contracts associated with the selected quote details.

Accept

The **Accept** button is used to accept the proposal contract associated with the quote detail. This button is enabled if the quote is `pending` or `complete` and at least one row in the quote details grid is selected.

When you accept a proposal contract, you are asked to supply the **Start Date** of the real contract that will be created.

Refer to Accepting / Declining Quote Details for more information about how accepting a quote detail causes a real contract to be created.

Decline

The **Decline** button is used to decline the proposal contract associated with the quote detail. This button is enabled if the quote is `pending` or `complete` and at least one row in the quote details grid is selected. When you decline a proposal contract, you are asked to supply a **Decline Reason**.

Refer to Accepting / Declining Quote Details for more information about declining a quote detail.

Cancel

The **Cancel** button is used to cancel the proposal contract associated with the quote detail. This button is enabled if the quote is `pending` or `complete` and at least one row in the quote details grid is selected.

Quote - Quote Details

This page is used to view a summary of the simulated bill segments linked to a quote detail.

Open this page using **Sales & Marketing, Quote** and navigate to the **Quote Details** tab to define this information. You can also display a quote detail by clicking the respective go to button in the quote details grid on the main tab.

Description of Page

Quote Info contains a concatenation of important information about the quote (e.g., its expiration date and status). **Quote ID** is the system-assigned unique identifier of the quote. These values only appear after the quote is added to the database.

The **Quote Details** scroll contains one entry for every quote detail linked to the quote (a summary of these quote details also appears on the Main tab). The remainder of this section describes the information displayed for each quote detail.

Copyright © 2024, Oracle and/or its affiliates.

> **Note: You don't have to use the scroll.** If the quote has many quote details, you don't have to use the scroll to work through them. Rather, you can use the go to button in the quote details grid on the main tab to position the respective quote detail in the following scroll. Also note, you can use Alt - and Alt + to scroll to the next and previous quote details in the scroll.

> **Note: You cannot maintain quote details on this page.** All maintenance functions (generate, delete, accept, decline and cancel) are performed on the **Main** tab.

Quote Detail ID is the system-assigned unique identifier of the quote detail.

Contract Info is a concatenation of the proposal contract's division, contract type, status and start date.

Contract Version is a snapshot of the proposal contract's version number at the time the quote detail was generated. This information is used by the system to detect if the proposal contract was modified after the quote detail was created.

Total Amount is the sum of the quote detail's simulated bill segments.

Create Date / Time is the date and time the quote detail was generated.

Remarks highlight special circumstances associated with the quote detail. Please see the description of this field on the main tab for more information.

The grid contains a row for each simulated bill segment linked to the quote detail. The following information appears in the grid:

- The first go to button transfers you to the bill segment page where the details of the simulated bill segment are shown.
- **Bill Segment Period** defines the start and end dates of the simulated bill segment.
- **Amount** contains the amount of the bill segment.
- **Remarks** highlight special circumstances associated with the bill segment.
- **Billing Scenario ID** allows you to go to the proposal Contract's billing scenario where you can tinker with product/charge type consumption and the bill segment period used to generate this bill segment. Refer to Proposal Contracts Contain Billing Scenarios for more information.

Quote - Quote Routings

This page is used to view the recipients of a quote. Refer to Printing Quotes for a discussion of how this information is used to produce quote images.

This page is also used to request a new copy of a quote.

> **Note: No routing information until completion.** A quote has no routing information until it is `complete`. At completion time, the system creates a routing detail for every person linked to the account that receives a copy of its quotes. Refer to Printing Quotes for more information about how this information is constructed.

Use **Sales & Marketing, Quote, Quote Routings** to open this page.

Description of Page

Quote Info contains a concatenation of important information about the quote (e.g., its expiration date and status). **Quote ID** is the system-assigned unique identifier of the quote. These values only appear after the quote is added to the database.

The **Quote Routing Information** scroll contains an entry for each recipient of the quote. This information is disabled after the quote has been routed to the recipient. The remainder of this discussion describes the fields that can be defined for each recipient of the quote.

Person ID identifies the person who was originally associated with the routing record. You may change the **Person ID** to any person in the system as this information is only used to default the recipient's name and address information, below.

The **Quote Routing Parameters** area contains information explaining how and when the quote is routed to the recipient. This information is gray when the quote routing information has been extracted or if a quote is reopened:

Copyright © 2024, Oracle and/or its affiliates.

- **Quote Route Type** is the method used to transmit the quote to the customer. This information defaults from Account - Person Information.
- **Batch Control** is the identifier of the batch of quotes in which the recipient's copy was downloaded. Embedded in this identifier is the batch control ID and the batch number. Note, this batch process is defined on the **Quote Route Type**.
- **Sequence** is the system-assigned identifier assigned to this quote routing.
- Turn on **Do Not Extract** if the quote should not be routed to the individual for whatever reason.
- **Reprint** is checked if an operator created this routing. This switch is populated by the system and is always protected.
- **Extract Date/Time** is the date and time the quote was downloaded. This information only appears after the download has commenced.

The **Mailing Address Information** area contains the recipient's name and address.

- If the recipient has an *override mailing name*, **Name1**, **Name2** and **Name3** contain the person's override mailing name. Otherwise, **Name1** contains the recipient's primary name (this name might also include a prefix or suffix if the related Account - Person relationship has these fields defined).
- The remaining fields are the recipient's address. The information that's displayed is dependent on the *Quote Route Type*.
  - If the recipient's **Quote Route Type** indicates a `postal` routing method is used, the address constituents are populated with the address specified on the *bill* address sourcethat is also specified on the account / person information. Note that the **Country** associated with the address controls the fields that appear in the address. Refer to Defining Countries for more information on the address constituents.
  - If the recipient's **Quote Route Type** indicates quotes are routed via *fax*, the recipient's **Fax ID** appears. The system knows which of a person's phone numbers is a fax number by the phone type.
  - If the recipient's **Quote Route Type** indicates quotes are routed via *email*, the recipient's **Email Address** appears.
- Please note the following in respect of the **Mailing Address Information**:
  - If the recipient's address is a mailing address (as opposed to fax or email), the **Country** associated with the address controls the fields that appear in the address. Refer to Defining Countries for more information on the address constituents.
  - If a user manually adds a routing record, the mailing address information defaults from the information on the current routing record.
  - If a user specifies a **Person ID** that is linked to the quote's account, the **Quote Route Type** defined for this person on the account is defaulted.
  - If a user specifies a **Person ID** that is *not* linked to the quote's account, the **Quote Route Type** on the *installation record* is defaulted. Note, if the quote route indicates the quote is routed via the postal service, the system will only be able to default the person's address if the person has an override mailing address on their correspondence information.
  - If the user changes the **Quote Route Type** after the initial default, the system populates the address information as described above.

Copyright © 2024, Oracle and/or its affiliates.

Chapter 11

Service Credits

Topics:

- The Big Picture Of Service Credit Membership
- The Big Picture of SC Events
- Maintaining Service Credit Memberships and Their Events

"Service credits" is a term used to describe a special program designed to reward a customer for service along with benefits related to such a program. For each program that a customer participates in, there is a service credit membership record. Service credit events are used to record the "points" earned and redeemed for the membership. The topics in this section provide further details related to the service credit functionality.

Fastpath: For information about designing and setting up your control tables for configuring service credit functionality, refer to Designing Your Service Credit Options.

Note: Service credits are optional. Service credits may not be set up in your system; they are only used if your company offers rewards programs to customers.

Note: Penalty Points. The service credits functionality is described in the documentation with the assumption that it is used for accumulating points to reward your customers. However, if your company has a business need to record penalty points for a customer, the service credits functionality may be used for that purpose as well.

The Big Picture Of Service Credit Membership

If a customer is participating in a rewards program, the customer has a service credit membership. The topics in this section provide background information about a variety of service credit membership features and functions.

Who are the Members?

Each membership has one or more "members". The members are defined by the accounts linked to the membership and by the financially responsible person(s) linked to each account.

A typical membership is linked to only one account. However, in some cases more than one account is linked to the membership. For example, a customer may have one account for banking service and one for insurance service, but both accounts are part of the same membership. In this example, one person with multiple accounts is linked to the membership, so there is only one member.

For each account linked to the membership, the service related to the account may be used to calculate the accumulated credits for the membership. Refer to *How Are Service Credits Earned?* for more information.

Note: An account may not be linked to more than one membership of the SAME membership type at a given point in time. In other words, an account may not have two United Airlines frequent flier memberships (otherwise, the account would accumulate double miles). However, an account may be linked to more than one membership if they are of different membership types. For example, an account may have a capital credits membership and a United Airlines frequent flier membership.

Note: Portal Preferences. Refer to Portal Preferences for more information.

Joint Membership

It is possible for more than one person to belong to the same membership. These persons may be linked to the same account or to different accounts. A typical scenario would be a married couple that shares in a membership. This is considered a "joint" membership.

The number of "members" for a membership is mainly for information purposes and typically has no bearing on the service credit amounts calculated for the membership. There may be legal restrictions or business practice restrictions controlling how you may set up your members. For example:

- In an airline's frequent flier program, a separate frequent flier account would exist for each customer. It is possible for you to create a frequent flier membership with more than one "member", but for the system, the important information is the external frequent flier miles number linked to the membership. The customer linked to this number accumulates the miles. For this case, your business practice may be that the person with the frequent flier account must be the only financially responsible person.
- For a capital credits membership, you may be required to distinguish between "single" memberships and "joint" memberships. If a single person gets married and would like the spouse to be added as a member, your practice may be to expire the single membership and to create a new "joint" membership to include both spouses.

Note: Co-op Members. Do not confuse the "members" of a co-op with the members linked to a service credit membership. At a cooperative, most customers are considered "members" of the co-op. However, in the system, separate membership records are created for each single or joint membership.

Changing the Membership Accounts

The link between the accounts and the membership is NOT effective dated. This means that if the list of accounts for a membership changes, you need to stop the membership and start a new one with the new list of accounts.

Copyright © 2024, Oracle and/or its affiliates.

> **Note: Stopping Memberships**. Stopping a membership does not necessarily prevent new service credits from accumulating. Refer to Stopping a Membership vs. Inactivating a Membership for more information.

Membership Application

Your business practice may dictate that the customer complete an enrollment form. Depending on your practice, the form may need to be received prior to starting membership or it may simply be a mechanism to gather supplemental information once the membership has begun.

Either way, you may consider setting up the system to handle forms as follows:

- Design your membership type to include a creation algorithm that generates a letter when the membership is created. The letter template could include the form to be filled out.
- Create the new membership in `pending` status. The letter and form are sent to the customer and the pending status indicates that you are waiting for the completed form. Refer to the Lifecycle of a Membership for more information.
- When the paperwork is received, change the status of the membership to `active`.

> **Note: Membership Events for Pending Memberships**. The system allows events to be created for `pending` memberships assuming that paperwork is supplemental information that should not prevent service credits from accumulating. If your business practice dictates that the paperwork must be received prior to accumulating service credits, the algorithm or background process that creates events for memberships of this type should be coded to only create events for `active` memberships. Refer to How Events Are Created for more information.

Creation of a Membership

Memberships may be created using the membership page when a customer has signed up for a special program. In addition, it may be your business practice to automatically create a membership under certain circumstances when a customer has signed up for service.

For example, if you are a co-op, perhaps every new customer is automatically enrolled in a capital credits membership. Or perhaps signing up for banking service automatically entitles your customers to membership in the free pay-per-view movies program. For these cases, the recommendation is to use the order page to start service for your customer and to use questions and miscellaneous fields to record information needed to set up the new membership correctly. Refer to Using Campaigns & Packages to Create a Membership.

Membership Fees

Some memberships may require a fee to be paid by the customer when joining. The behavior of this fee may differ depending on whether it is refundable or not.

- If the fee is nonrefundable, you create an contract with a simple adjustment or a one time billable charge to levy the fee. Refer to Defining Control Tables for a Nonrefundable Fee.
- If the fee is refundable, in addition to defining the contract to hold the fee, you need to put code in place to initiate the refund. Refer to Setting Up a Refundable Fee for more information.

Stopping a Membership vs. Inactivating a Membership

You may stop a membership by entering an end date for the membership. However, depending on how events are created for your membership, stopping the membership may not prevent new events from being created. For example:

- If you stop a frequent flier membership on May 15 th and a bill is subsequently created for the period May 1 st through June 1 st, your business rules may dictate that frequent flier miles should be accumulated for this bill because the membership was active for part of the period covered by the bill.

Copyright © 2024, Oracle and/or its affiliates.

- For a capital credits membership, capital credit allocations are calculated for the prior year. If a membership stopped in the middle of the previous year, the membership should still get capital credits for the part of the year that the membership was still in effect.
- Capital credit retirements are calculated based on capital credit allocations accumulated in the past. If a membership has been expired, the members are still eligible for retirement amounts for the accumulated capital credit allocations. These retirements may continue for many years after a membership is stopped.

Note: Membership Status. For all the above scenarios, the system assumes that the status of the membership remains `active`.

You may have some circumstances where you do not want new events to accumulate after a membership is stopped. To prevent new events from being created, you must change the status of the membership to `inactive`.

If a customer moves away or otherwise stops service with your company, your business rules dictate whether or not the membership should be stopped or changed to `inactive`. The following issues must be considered:

- Will service credit events still get created for these memberships? If so, is that acceptable?
  - For frequent flier memberships, the miles are calculated based on new bill amounts for the membership's contracts. If the contracts that contribute to the membership are expired, no new bills are generated so no new miles are accumulated.
  - For capital credits, after a customer moves away, the capital credit allocation for the final year of the customer's service may still need to be calculated. In addition, this customer is eligible for the capital credit retirement amounts, even though they are no longer a customer.
- What happens if the customer returns to your company and signs on for service again? Assuming that the same account number is used, is it ok for the membership to be automatically reinstated? Or should the customer reapply? If the customer should reapply, you may consider updating the membership with a stop date when the customer originally stopped service.

Note: There is nothing in the system logic that automatically updates the membership stop date or the status when contracts that contribute to the membership are stopped.

The Big Picture of SC Events

The service credit events record the accumulation and redemption of awards earned by the membership. The topics in this section provide background information about a variety of service credit event features and functions.

Note: Events. For convenience, this document often refers to service credit events simply as "events".

Event Amounts May Contribute to a Balance

The event amount may be positive or negative, based on whether the event is accumulating points earned for the membership or redeeming points for the membership. For some types of memberships, these credit and debit amounts should be added together to produce the overall balance for the membership.

For some types of memberships, a balance is not applicable. This is true for memberships where the redemption of the points does not occur within the system. For example, if the membership accumulates frequent flier miles on behalf of an airline, the customer redeems the earned miles with the airline, not with your company. As a result, there is no need to calculate a balance for the membership.

If you want the system to calculate a balance, you must set a switch on the membership type page. For memberships that calculate a balance, the balance may be viewed on the membership page.

Copyright © 2024, Oracle and/or its affiliates.

Events May Indicate a Fiscal Year

For some types of memberships, the amount of each event is associated with a specific fiscal year. This is true for capital credit memberships where capital credit allocations are calculated for a fiscal year. When the capital credits are redeemed (also called retired), the retirement events must also indicate a fiscal year. Because all events for this type of membership are related to a fiscal year, the system can calculate the balance for the events by fiscal year.

If you want the system to require a fiscal year for every event in the membership, you must set a switch on the membership type page. For memberships that calculate a balance and indicate a fiscal year, the balance for each fiscal year may be viewed on the membership page.

Events May Indicate a Subcategory

For some types of memberships, the amount of each event is further grouped by a subcategory. This is common for capital credit memberships where multiple capital credit allocations are calculated. For example, each year, you may calculate allocations for generation and for transmission. Events for memberships of this type must indicate a subcategory.

Cooperatives that allocate capital credits for more than one subcategory, typically want to view the balance of the events over time broken down into these subcategories. When the capital credits are redeemed or retired, the retirement events must also indicate a subcategory. Because all events for this type of membership are related to a subcategory, the system can calculate the balance for the events by subcategory.

Note: Partial Retirement. For some cooperatives, it is common that certain capital credit allocations are never retired. Refer to Partial Retirement for more information.

If you want the system to require a subcategory for every event in the membership, you must define the valid subcategories on the membership type page. For memberships that calculate a balance and indicate a subcategory, the balance for each subcategory may be viewed on the membership page.

Note: Calculating Subcategory Balances. If events for your membership indicate a subcategory and fiscal year, balances for the membership are calculated for both groupings.

An Event May Cause Other Actions to Occur

When an event is generated, there are often other actions that need to take place. For example:

- If an event is created that allocates points for the customer, you may want to generate a temporary message on the customer's next bill.
- If a capital credit allocation event is created, the general ledger is affected so an adjustment must be created to post to the general ledger. Refer to Financial Transactions for an Event for more information.
- If an event is created to add frequent flier miles, the event must be marked for extraction so that the information is sent to the appropriate airline. Refer to Interface Membership Information to a Third Party for more information.
- If an event to retire capital credits is created, the customer's balance is affected so an adjustment must be created to transfer the capital credit to the members' contracts. Refer to Financial Transactions for an Event for more information.

The system relies on algorithms linked to the event type to perform the actions described above and any other action that may be required by your business.

Fastpath: Refer to Service Credit Event Types for more informationabout the algorithms provided with the base product.

Note: Event Cancellation Algorithms. The event type also defines algorithms to be executed when an event is canceled.

Copyright © 2024, Oracle and/or its affiliates.

Financial Transactions for an Event

An event may be related to financial transactions. There are two types of financial transactions that may be linked to the event:

- Financial transactions that `contributed to` the event. These are bills, payments or adjustments that caused the event to be created. For example, if you generate an event for frequent flier miles as a result of some amount billed for the customer, the bill segments associated with the billed amount are linked to the event. The FTs are linked to the event for audit purposes.
- Financial transactions that are `created by` the event. In an event may cause other actions to occur, we saw examples where an event completion algorithm may cause adjustments to be created. The FT ID for these adjustments is linked to the event as audit.

Note: Automatically Canceling Event Financial Transaction. The `created by` financial transactions may get canceled if the event is canceled AND you have plugged in an appropriate Event Cancellation Algorithm.

For financial transactions that are created by the event, the system must post the financial transactions to one or more appropriate contracts. The contract to use depends on your business rules. For example:

- If the event affects the balance of accounts linked to the membership, the event completion algorithm may create adjustments for contracts that contribute to the membership.
- If the event affects the general ledger and does not affect any balance for the account, the event completion algorithm simply needs a contract to use for the resulting adjustment. Rather than arbitrarily picking any contract in this case, a special miscellaneous contract should be linked directly to the membership for such adjustments.

How Events Are Created

Although a user may create an event manually on the service credit event page, most events are created automatically by an algorithm or by a background process. The following topics further describe these options.

Events Created by an Algorithm

You would design an algorithm to create your events if the event should be triggered by an occurrence in the system, such as starting service or generating a bill. For example, perhaps every time you are billed for banking and/or insurance service, the system adds one frequent flier mile for every $10 spent on service. Refer to *How Are Service Credits Earned?* for more information.

The base product provides a bill completion algorithm to create an event as a result of the amount billed for contracts linked to the membership. Refer to CBCM-SC for more information.

Events Created as a Result of a Background Process

You would use a background process to create the event if the calculation is done periodically or on an ad-hoc basis and does not get triggered by something occurring in the system. For example, the capital credit allocation and retirement processes are typically run once a year, when the business has determined that they are ready to calculate these amounts.

Fastpath: Refer to Allocating Capital Credits and Capital Credit Retirement for more information.

Creating Pending Events

Events may be created in pending or complete status. Only complete events cause the completion event algorithms to run and cause other actions to occur. You may use the status of `pending` if you want to review the calculated amount prior to completing the event.

For example, when a cooperative runs the background process to allocate capital credits, it may create the new events in the `pending` status. This allows the co-op to review the total amount of capital credits calculated for all members.

Copyright © 2024, Oracle and/or its affiliates.

If this total is in line with the co-op's expectations, they run the background process again creating the new events in `complete` status. Only at this point do the completion event algorithms run.

Events May Be Transferred

In some situations, you may want to transfer service credit events from one membership to another on the service credit event page. For example, if a member dies, you may decide to transfer earned service credits to the membership for a beneficiary. When transferring a service credit:

- The target membership must be of the same membership type as the original membership.
- The original service credit event is canceled and any event cancellation algorithms linked to the SC event type are executed.
- A new service credit event is created and completed and any event completion algorithms linked to the SC event type are executed.
- The new service credit event references the original event ID in the **Transferred From** event field.

Note: `Contributed to` FTs linked to the original event are not transferred to the new event. If you have `created by` FTs linked to the original event, the recommendation is to ensure that an event cancellation algorithm cancels this FT and that an event completion algorithm for the new event creates an appropriate new FT for the new membership.

CPCRALOC - Allocating Capital Credits

The base product provides a background process that allocates capital credits to members of a cooperative. The result of this process is the creation of new service credit events for capital credit memberships. This background process is typically run after the profits for the previous year are calculated. During that time, this job could be run a number of times for any of the following reasons:

- At the beginning of the allocation process, you may know the total amount to allocate to all members; however, you may estimate the allocation factor that determines the individual amounts that get distributed to the members. You may need to change your estimate and rerun the process until all the individual allocations add up to the desired total amount. To run the job with the estimated allocation factors, indicate that you want events created as pending. When you confirm the correct allocation factor, a final run should be executed with an indication that the events should be created as complete. Every time the process runs, it will first remove any pending events for the same input event type, subcategory and fiscal year. Refer to Creating Pending Events for more information.
- If the cooperative has more than one subcategory, the process must be run for each subcategory. In fact, the above trial-and-error process may need to be followed for each subcategory.

You must provide the service credit event type to use for the new events. This process determines the membership type related to the event type. The process then finds all pending and active memberships for this membership type. To calculate the capital credit allocation, the system finds all applicable bill segments for each contract that contributes to the membership.

Capital credit allocations are calculated for a specific fiscal year. The fiscal year is input to the background process and is stamped on any new events created by the process. The system also uses the fiscal year to determine the applicable bill segments for the membership's contracts. It uses the bill segment's accounting date and the accounting calendar to determine the fiscal year for each bill segment.

Note: Canceled Bill Segments. Any bill segments that were canceled in the fiscal year are included as well.

Note: Accounting Calendar. This process assumes that only one accounting calendar exists for the company. This is the case for companies that operate in a single jurisdiction, which is true for cooperatives.

Note: Allocation Period. If the membership was only effective for part of the fiscal year, the allocations are only calculated for the period of time that the membership was in effect.

Copyright © 2024, Oracle and/or its affiliates.

For each applicable bill segment, the system determines the SQ total and the total sales amount for each segment. To calculate the totals correctly, a characteristic type and value to identify SQ bill lines must be provided to the background process and a characteristic type and value to identify bill lines contributing to the total sales amount must be provided to the background process.

Fastpath: Refer to Identifying SQ and Sales Information for Historical Bill Segments for more information.

If your membership type indicates subcategories to be used for the events, you must provide a subcategory as input. If you have multiple subcategories for your membership, this process must be run once for each subcategory.

The calculation of the capital credit allocation amount typically uses an allocation factor. This factor is often different for each subcategory. You will probably define a different bill factor for each subcategory and use the appropriate bill factor when submitting the process for each subcategory.

This process also assumes that the allocation factor differs by the customer's revenue class. It assumes that the bill factor's characteristic type is revenue class. Refer to Designing Bill Factors for Credit Allocation for more information. For each contract linked to the membership, the appropriate allocation factor is determined based on the revenue class for the contract type.

Capital credits for the input subcategory are calculated as follows:

Fiscal year sales amount - (fiscal year SQ amount * revenue class allocation factor) = Allocation Amount

The calculated amount is compared against a minimum amount, input to the background process. If the calculated amount is less than the minimum, no event is stored for this amount.

After determining the amount, a new service credit event is created for the membership. The event is stamped with the input subcategory, service credit event type and fiscal year.

The FTs for all the bill segments used to determine the allocation amount are linked to the event as `contributed to` FTs. Refer to Financial Transactions for an Event for more information.

Note: Rerunnable Process. This process is restartable and rerunnable. When running the process, for each membership the system checks whether there is already a `complete` event for the input service credit event type, subcategory and fiscal year. If so, it does not create a new event.

Fastpath: Refer to CPCRALOC for more information.

CPCRRETR - Capital Credit Retirement

Note: Calculation Method. There are various calculations that different cooperatives may use when determining the amount to retire. The base product has provided one calculation method.

This background process is run when a cooperative is ready to pay out allocated capital credits to its members. This process is referred to as retirement. The result of this process is the creation of new service credit events for capital credit memberships. Capital credits are paid as the cooperative's financial position permits. During that time, this job could be run a number of times for any of the following reasons:

- You may choose to run the process and create pending events to review the results prior to finalizing the retirement. When you are happy with the results, a final run should be executed with an indication that the events should be created as complete. Every time the process runs, it will first remove any pending events for the same input event type and subcategory.
- If the cooperative has more than one subcategory, the process may be run separately for each subcategory. You may also run the process with no subcategory indicated and then all subcategories are processed.

Note: Partial Retirement. For some cooperatives there may be capital credits for some allocated amounts that are never retired. If that is the case, simply run this process for each subcategory whose amounts do get retired.

Copyright © 2024, Oracle and/or its affiliates.

You must provide the service credit event type to use for the new events. This process determines the membership type related to the event type. The process then finds all pending and active memberships for this membership type.

This process receives a percentage to retire as input. It determines the current balance of all completed events for the subcategory. The retirement amount for the subcategory is calculated as the current balance * input percentage to retire. The process then determines the outstanding balance for each subcategory and fiscal year combination. For every combination that still has a balance, the process creates events to relieve the outstanding balance for the oldest fiscal year first. It creates events for subsequent fiscal years until the full retirement amount is used.

In order for the process to be rerunnable and restartable, a Control ID is used to mark retirement events that are created by the process. The Control ID is input to the routine and is determined by the user. When the process runs, it determines whether an event already exists for the subcategory / Control ID combination. If so, new events are not created for that subcategory. If you run the process multiple times for the reasons described above, you should use the same Control ID each time. This way the system knows that the process is being run again for the same retirement calculation.

Fastpath: Refer to CPCRRETR for more information.

Maintaining Service Credit Memberships and Their Events

The topics in this section describe how to maintain a service credit membership and its events.

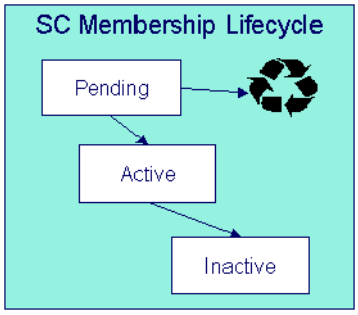
Fastpath: Refer to The Big Picture Of Service Credits for background information about service credits.

Service Credit Membership

The topics in this section describe how to maintain service credit memberships. Refer to The Big Picture Of Service Credit Membership for background information about memberships.

Lifecycle of a Membership

The following diagram shows the possible lifecycle of a service credit membership.



`Pending` A service credit membership may start out in the `Pending` state. A membership is typically created in this state if the creation of the membership requires approval or any manual step (e.g. waiting for paperwork) that needs to be completed prior to activation.

Note: Events for Pending Memberships. Service credit events may still be created for membership records in the `Pending` state.

`Active` A service credit membership is in the `Active` state when a customer is actively participating in a service credit membership. A membership may be created in the `Active` state or it may transition to the `Active` state from `Pending` when all the necessary approval or paperwork is complete.

Note: Manual Activation Only. There is no system functionality that automatically changes the status of a membership from `Pending` to `Active`. This is a manual step.

Copyright © 2024, Oracle and/or its affiliates.

`Inactive` If a customer is no longer actively participating in a service credit membership, change the status to `Inactive`. This will prevent any new events from being created for the membership.

Service Credit Membership - Main

This page is used to maintain and display general information about a service credit membership. Open this page using **Customer Management, Service Credit Membership**.

Description of Page

Membership Info and **Membership ID** are displayed on every page. These values only appear after the membership exists on the database. The **ID** is a system-assigned random number. The **Membership Info** is a concatenation of important details about the membership.

Membership Status defines the state of the membership.

Fastpath: Refer to Lifecycle of a Membership for more information about membership status.

Click the **Activate SC** button to activate a `Pending` membership. When this button is clicked, the system executes any Membership Activation algorithms that are linked to the service credit membership type.

Click the **Inactivate SC** button to transition an `Active` membership to `Inactive`. When this button is clicked, you are prompted to provide an **Inactive Reason** and the system executes any Membership Inactivation algorithms that are linked to the service credit membership type.

Indicate the **SC Membership Type**. This field is important as it controls numerous aspects of the behavior of the membership and its events.

Fastpath: For more information about what the membership type controls, refer to Setting Up Service Credit Membership Types.

Indicate the **Start Date** and **End Date**, which define the period for which this membership is effective. Leave the end date blank if this membership is effective indefinitely. Refer to Stopping a Membership vs. Inactivating a Membership for more information.

If this membership is related to a record in an external system, use the **External ID** to indicate the unique identifier of the record in the external system.

If events linked to your membership may generate miscellaneous adjustments, indicate the **Contract** to use for these miscellaneous transactions. The account for this contract must be one of the accounts linked to the membership.

The **Events** grid displays the events linked to the membership. To view any of the events use the drill down button.

If your SCM type indicates that the membership has a balance, the tree at the bottom displays information about the balance of the membership based on the amounts for `completed` events. If the SCM Type indicates subcategories and/or a fiscal year is required, subtotals for `completed` events are also displayed for each subcategory / fiscal year and by fiscal year / subcategory, if applicable. If any `pending` events exist, the tree includes a node with information about these events.

Service Credit Membership - Account

This page is used to maintain the accounts linked to a service credit membership. Open this page using **Customer Management, Service Credit Membership** and navigate to the **Account** tab.

Description of Page

Indicate the **Account ID** for each account linked to this membership. Typically a membership will be linked to only one account. However, if a person has multiple accounts and the service credits should span the accounts, multiple accounts are linked to the same membership.

Copyright © 2024, Oracle and/or its affiliates.

Service Credit Membership - Characteristics

The characteristics page contains information that describes miscellaneous information about the service credit membership. Open this page using **Customer Management, Service Credit Membership** and navigate to the **Characteristics** tab.

Description of Page

> **Note: Characteristic Types.** You can only choose characteristic types defined as permissible on the service credit membership record. Refer to Setting Up Characteristic Types & Their Values for more information.

To modify a characteristic, simply move to a field and change its value. To add a new characteristic, click + to insert a row then fill in the information for each field. The following fields display:

Effective Date Define the date on which the characteristic becomes effective. The effective date defaults from the Installation Record .

Characteristic Type Indicate the type of characteristic.

Characteristic Value Indicate the value of the characteristic.

> **Note:** A service credit membership's characteristics default from the service credit membership type.

Service Credit Membership - Membership Tree

The membership tree page displays accounts, contracts and persons linked to the service credit membership. Open this page using **Customer Management, Service Credit Membership** and navigate to the **Membership Tree** tab.

Description of Page

The tree displays each account linked to the membership. For each account, a node exists for:

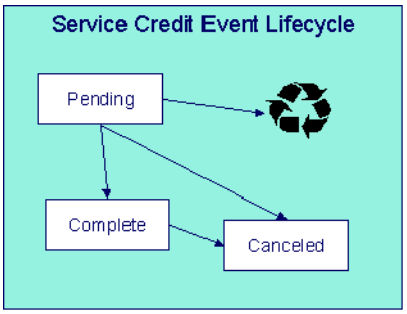
- Each contract whose contract type is included in the service credit membership type's contract type collection.
- Each financially responsible person linked to the account. These persons are considered the members.

Maintaining Service Credit Events

Service credit events record credits earned or redeemed for a given membership. The topics in this section describe how to maintain service credit events. Refer to The Big Picture of Service Credit Events for background information about memberships and their events.

Lifecycle of a Service Credit Event

The following diagram shows the possible lifecycle of a service credit event.



`Pending` A service credit event may start out in the `Pending` state. An event is typically created in this state when a review of the event is desired before completing it. Refer to Creating Pending Events for more information.

`Complete` A service credit event is typically created in the `Complete` state. When an event is completed, the system executes any event completion algorithms indicated on the service credit event type.

Copyright © 2024, Oracle and/or its affiliates.

`Canceled` If a service credit event is no longer valid, it may be `Canceled`. When an event is canceled, the system executes any event cancellation algorithms indicated on the service credit event type.

Service Credit Event - Main

Open this page using **Customer Management, Service Credit Event**.

Description of Page

The **Service Credit Event Info** is a concatenation of important details about the service credit event. The **Service Credit Event ID** is a unique system-assigned random number. These values only appear after the service credit event exists on the database.

Service Credit Event Status defines the state of the service credit event.

> **Fastpath:** Refer to Lifecycle of a Service Credit Event for more information.

Click the **Complete** button to change a `Pending` event to `Complete`. When this button is clicked, the system executes any event completion algorithms that are linked to the service credit event type.

Click the **Transfer** button to transfer this event to another membership. When this button is clicked, you are prompted to provide the membership ID to transfer the event to. The new membership must have the same membership type as the original membership. Refer to Events May Be Transferred for more information.

Click the **Cancel** button to cancel this event. When this button is clicked, the system executes any event cancellation algorithms that are linked to the service credit event type.

The **Create Date/Time** captures the date and time that this event was created. This defaults to the system date/time.

Indicate the service credit **Membership ID** to which this event should be linked.

Indicate the **Service Credit Event Type**. This field may define algorithms to be executed when the event is completed or canceled. Refer to Setting Up Service Credit Event Types for more information.

If your service credit membership type indicates one or more subcategories to be associated with events for this type of membership, indicate the appropriate **Subcategory Name** for this event.

If your service credit membership type indicates that fiscal year is required, indicate the appropriate **Fiscal Year** for this event.

Enter the **Service Credit Event Amount**. If your service credit membership type indicates that your membership has a balance, this amount is used to calculate the overall balance for the membership as well as a balance for the subcategory and/or fiscal year, if applicable.

If this event was created as a result of a transfer from another membership, the **Transferred From Event** ID is displayed.

If the information about this event may be downloaded to a third party system, you should have an event completion algorithm that stamps the appropriate **Batch Control** and **Batch Number** onto the event.

The FT collection displays information about financial transactions related to this event. Two different categories of financial transactions may be linked to the event.

- FTs that `contributed to` this event. These financial transactions played a part in the creation of this event. For example, if frequent flier miles are earned as a result of newly created bill segments, the FTs related to these bill segments are linked to the event and marked as `contributed to` FTs. This information is purely for audit purposes. It is the responsibility of the algorithm or background process that creates the event to populate the collection of `contributed to` FTs. Refer to How Events Are Created for more information.

> **Note:** The amount of the `contributed to` FTs will not necessarily add up to the amount of the event. Refer to the algorithm or background process that calculates the amount of the event to understand how the event amount is calculated based on the amount of the `contributed to` FTs

- FTs that were `created by` this event. These financial transactions are created as a result of this event. For example, if you have generated a capital credit allocation event, perhaps you need to post information to the general ledger. An allocation event should therefore generate an adjustment to post the effect on the GL. The adjustment's FT is

Copyright © 2024, Oracle and/or its affiliates.

linked to the event as an audit. In order to create a financial transaction related to this event, you must plug in an appropriate event completion algorithm on your service credit event type.

Fastpath: Refer to An Event May Cause Other Actions to Occur for more information.

Copyright © 2024, Oracle and/or its affiliates.

Chapter 12

Loans

Topics:

- The Big Picture Of Loans
- Loan Maintenance
- How To

There are many reasons why your company may loan money to customers.

> **Note: Loans are optional.** You need only set up loans if your organization loans money to customers. For information on setting up the system to enable loans, refer to Defining Loan Options.

The Big Picture Of Loans

The topics in this section provide background information about loans.

Loans Are Contracts

Loans are initiated by creating a loan contract for a customer. The loan contract (and its contract type) contains the loan's terms, including the loan amount, periodic payment amount, the number of amortization periods, the interest rate, and the periodicity of the bills. Loan contracts are just like other contracts in many ways:

- The system bills the customer for the periodic payment amount using information on the contract.
- The customer pays the billed amount.
- If the customer doesn't pay, credit and collections may start a collection process to encourage the customer to pay (depending upon how the loan debt class has been set up on your system).

Loan contracts differ from other contracts in the following ways:

- Loans have interest charges.
- Loans have amortization schedules.
- Loan contracts are maintained on a separate transaction (refer to *Loan - Main* for more information).

Note: Loan contracts are created using Start / Stop. Loan contracts are created using start / stop just like all other contracts. The start/stop transaction has special loan functionality that allows an operator to specify the contract-specific loan terms described above.

Loan Interest Charges

When a bill is generated for a loan, the interest charge is calculated using an algorithm. The base package loan interest charge algorithm type calculates simple interest based on:

- Unbilled principal (i.e., the payoff balance minus the current balance)
- The number of billing periods covered by the bill
- The interest rate

Note: No interest on interest and no interest on past due amounts. Just like a classic home loan, the base package algorithm does not calculate interest on interest, nor does it calculate interest on past due amounts. However, your system may be set up to levy a late payment charge.

After the interest charge is calculated, the bill segment FT algorithm generates the financial transactions for the bill. The base package FT algorithm generates a financial transaction that adds the interest charge to the payoff balance. In other words, if the customer wanted to pay off the loan, they would owe the principal balance plus the interest charge.

Fastpath: For more information, refer to Billing For Loans And Interest Calculation.

Loan Amortization Schedules

A loan's periodic payment amount is calculated such that the customer repays the loan in equal amounts over the term of the loan (with the exception of the first and last bill segments, which are typically for different amounts). The amortization schedule is a projection of the amount of principal and interest in each payment over the life of the loan. The amortization schedule may change, for example if the interest rate changes or the customer makes an overpayment (reducing the principal balance).

Copyright © 2024, Oracle and/or its affiliates.

The amortization schedule is a theoretical projection of the future. The term theoretical is used because the customer's actual payments can cause the schedule to change. For example, if a customer overpays a bill, the loan's length decreases and the amount of principal and interest in each scheduled payment changes.

A loan's amortization schedule is calculated when you click the **Calculate** button on *Loan - Main*. When this button is clicked, an algorithm plugged in on the loan contract type actually calculates the amortization schedule.

You can view a loan's latest amortization schedule using *Loan - Main*. You can also change the loan's payment periods or periodic payment amount on this page and recalculate the loan's amortization schedule.

Negative Amortization

The system does not allow you to use a periodic payment amount that is less than the periodic interest amount. If it did, the customer's payoff balance would increase over time (this is referred to as "negative amortization"). There are two methods you can use to specify a payment amount in the system. Depending on which method you use, the system informs you of negative amortization differently:

- If you enter a payment amount that is too small on the Start confirmation page or in the **Renegotiate Future Payments** grid on Loan maintenance, you get an error message when you click the **Calculate** button. You must fix the payment amount before you can start the loan or save your changes.
- If you manually override the **Payment Amount** on Loan maintenance, you can save a payment amount that is too small to cover the interest charges. However, the system notifies you that it is not able to calculate the **Forecasted Schedule** because of negative amortization. Additionally, bill segments for the loan contract are produced in error until the payment amount is adjusted appropriately.

Loan Overpayments

Overpayments reduce the principal amount (the amount owed on the loan), which follows the philosophy adopted by a typical home loan. When the payment is made, any overpayments are distributed according to the overpayment distribution algorithm defined for the customer class. Typically, loan contracts should be set up so that the overpayment is applied to the principal balance.

When the payment transaction is frozen, the system checks to see if there is a credit amount on the loan contract's current balance. If a credit exists, the customer has made an overpayment and an adjustment is created to transfer the amount of the credit from the contract's current balance to its payoff balance, thus reducing the principal by the amount of the overpayment.

Fastpath: For more information about the adjustments created for loan overpayments, refer to Overpayments On Loans.

Note: Overpayments and interest. The base package interest calculation algorithm (plugged in on the loan's contract type) does not take into consideration the exact date that the overpayment is made when calculating the interest for the period. It only takes into consideration the outstanding principal amount (payoff balance - current balance) at the time of the interest calculation.

The Lifecycle Of A Loan

Loans have the same lifecycle as other contracts (e.g., `pending start`, `active`, `pending stop`). You should rarely have to manually change a loan contract's status. Why? Just like other contracts, the system changes the status behind-the-scenes when various events take place. For example, loans in the `pending start` state are automatically activated by a system background process.

If you need to manually change a loan contract's status (e.g., you need to cancel a loan that was inadvertently started), you can do so from Contract - Main.

Copyright © 2024, Oracle and/or its affiliates.

Finalizing And Closing A Loan

A loan contract is automatically stopped when the last bill segment is generated for the loan. If the remaining principal balance (plus accrued interest) is less than or equal to the loan payment amount, the contract status is changed to `pending stop` and the bill segment is marked as the closing bill segment. The contract can then be `stopped` by the user (manually using the Contract page) or by the contract activation background process. When the customer pays the last bill, the loan contract is closed.

Navigating To The Loan Maintenance Page

Any of the following methods can be used to navigate to the *Loan Maintenance* transaction:

- A user can set up a favorite link associated with the loan transaction.
- Via any of the following menus:
  - Account context menu
  - Contract context menu
  - Main menu

Note: Alerts and Loans. If the loan contract type is set up with an alert, the alert zone in the dashboard indicates whether the loan exists for the current account. You are transferred to the contract maintenance transaction (not the loan maintenance transaction) if you drill down on the alert.

Loan Maintenance

You can use loan maintenance to view the loan's amortization schedule and financial history as well as to change the terms of a loan, including the loan amount (prior to contract activation only), interest rate, payment amount and number of payment periods.

Note: Starting Loans. You cannot start a loan from Loan - Main; you must use Start / Stop. Refer to How To Start A Loan Contract for information.

Open **Customer Management, Loan** to maintain a loan.

Description of Page

The fields at the top of the page are informational only. The **Payoff Balance** is the total amount owed by the customer (excluding any interest that may have accrued since the last bill). The **Current Balance** is the amount currently owed by the customer. The **Principal Balance** is the total principal amount left unbilled. Refer to Payoff Balance and Current Balance for Loans for more information.

The **Loan Amount** is the amount originally loaned to the customer. The **Standard Interest Rate** is the standard interest rate defined on the loan's contract type. If the contract type allows, the standard interest rate may be overridden manually. The **Next Payment Amount** is the effective periodic payment amount as of the next bill.

The **Payment(s)** shows the number of payments that were used to calculate the original loan schedule.

The **Override Interest Rate** grid is used to override the standard interest rate. The interest rate can only be modified if the loan's contract type allows it. Note that changing the interest rate changes the loan amortization schedule. To override the interest rate:

- Specify an **End Date** for the override rate in effect if there is one.
- Click + and specify the **Start Date** and the new **Interest Rate**. You can also specify the **End Date** of the new override rate, but this is not required.

The **Payment Amount** grid is used to maintain the effective periodic payment amount. Note that changing the payment amount changes the amortization schedule. To change the payment amount:

Copyright © 2024, Oracle and/or its affiliates.

- Click + and specify the **Effective Date** and the new payment **Amount**.

> **Note: Entering a payment amount that is too small.** If you save a payment amount that is too small to cover the periodic interest charges, this causes negative amortization. The system is not able to recalculate the amortization schedule and bill segments are produced in error until the problem is corrected.

The **History** grid shows the loan's financial history in reverse chronological order. The financial history grid displays the following information:

- The **Arrears Date** is used to compute how many days old the debt is. In the context of a loan, debt only refers to the billed amount (the amount the customer needs to pay to stay current).
- **Financial Transaction Type** displays the type of financial transaction except for adjustments. For adjustments, the adjustment type's description is shown.
- **Current Amount** is the effect of the FT on the contract's current balance.
- **Current Balance** is the amount owed by the customer at the time of the transaction.
- **Payoff Amount** is the effect of the FT on the contract's payoff balance. The value is grayed out if it is the same as the current amount.
- **Payoff Balance** is the total amount owed by the customer. The value is grayed out if it is the same as the current balance.

> **CAUTION:** If you do not understand the difference between payoff balance and current balance, refer to Payoff Balance And Current Balance For Loans.

The **Forecasted Schedule** displays the loan amortization schedule. The first row of the schedule shows the current cumulative interest, current cumulative principal and the current principal balance. All other rows show the future amortization schedule based on the loan details. The forecasted schedule includes the following information:

- The **Loan Period** displays the forecasted payment period dates based on the billing frequency and the payment periodicity. The Loan Period is blank for the first row, which shows current cumulative amounts.
- The **Amount** displays the projected amount of that payment. The Amount is zero for the first row, which shows current cumulative amounts only.
- The **Interest** displays the projected amount of interest in the payment amount. The Amount is zero for the first row, which shows current cumulative amounts only.
- The **Principal** displays the projected amount of principal in the payment amount. The Amount is zero for the first row, which shows current cumulative amounts only.
- The **Cumulative Interest** is the total interest amount that would theoretically be billed at that loan period.
- The **Cumulative Principal** is the total principal amount that would theoretically be billed at that loan period.
- The **Principal Balance** is the remaining principal balance that would theoretically be remaining at the loan period.

> **Note: Forecasted schedule.** The loan schedule algorithm calculates the values in the forecasted schedule. Depending on the algorithm set up for your installation, the values may have a different meaning.

The **Renegotiate Future Payments** area allows you to change the terms of the loan. You can modify the following options:

- You can modify the **Loan Amount** only if the contract has not been activated. After the contract is activated, this field is for information only.
- The **Interest Rate** allows you to change the interest rate. When changing the interest rate, specify either the number of payment amounts or the payment amount and click Calculate to recalculate the amortization schedule.
- The **Number of Payment Periods** allows you to change the number of periods remaining in the loan's term. The **Frequency** displays the periodicity of the payments. If you change the number of payment periods, use the Calculate button to recalculate the payment amount.
- The **Payment Amount** allows you to change the amount of individual payments. If you change the payment amount, use the Calculate button to recalculate the number of periods.
- The **Calculate** button recalculates the payment amount (when you specify the interest rate and the number of payment periods) or the number of payment periods (when you specify the interest rate and the payment amount) and recomputes the loan amortization schedule based on these new terms.

Copyright © 2024, Oracle and/or its affiliates.

Note: Implementing New Loan Terms. The changes that you make using **Override Interest Rate**, **Payment Amount** and **Renegotiate Future Payments** do not take effect until you save the loan. Changes made to the **Renegotiate Future Payments** section are only valid for save after you click Calculate, and they override any other changes.

Note: Saving Renegotiated Future Payments. When you save a loan that has renegotiated payments, the system may add new records for the override interest rate, payment amount, or both. It uses the start date of the next loan period as the start date of the new interest rate and/or as the effective date of the new payment amount. If the loan has not yet been billed, the system updates the interest rate and/or payment amount created when the loan was started.

How To

The topics in this section describe how to perform common loan functions.

How To Start A Loan Contract

Loan contracts are created using start / stop just like all other contracts. The start/stop transaction has special loan functionality that allows an operator to specify the contract-specific loan terms described above.

Creating a loan contract involves the following steps:

- Search for the account for which you want to create a loan contract in the Start/Stop screen.
- Enter a **Start Date** and a **Start Method** of `Start a Contract`.
- Indicate the **Division**/ **Contract Type** of the loan contract and click the **Start** button.
- After you click **Start**, the Start Confirmation page appears. When you select a loan contract type, the Start Confirmation page displays special fields that allow you to specify the loan's terms.
- Enter the **Loan Amount** (the amount that the customer is borrowing). Note that the label for the Total Amount To Bill (e.g., Loan Amount) may appear differently depending on how the contract type was defined.
- The system can automatically calculate either the payment amount (based on the loan amount and the payment periods) or the payment periods (based on the loan amount and the payment amount). Enter either the **Payment Amount** or the **Payment Periods** and click **Calculate**. Note that the label for the Recurring Charge (e.g., Payment Amount) may appear differently depending on how the contract type was defined.
- Once the loan parameters are populated, click **Start**.

The loan is created in the `pending start` state. The next time the contract activation process runs, it will activate the loan contract. If you want to activate the loan immediately, navigate to the contract page and click **Activate Contract**.

Note: Overriding the Interest Rate. The **Interest Rate** is a display only field. The interest rate is taken from the value of the interest bill factor defined on the contract type. After starting a loan, you can manually override the interest rate using *Loan - Main* if the contract type allows it.

Note: Changing The Loan Amount. You can use loan maintenance to change the loan amount for an contract as long as the contract has not yet been activated. It may be helpful to transfer to *Loan - Main*, and verify the amortization schedule before activating the contract.

How To Modify The Terms Of A Loan

After a loan is active, you may want to modify its terms. You can change number of payment periods, the payment amount, and the interest rate (if allowed by the contract type). To modify the terms of a loan:

- Navigate to the Loan maintenance page.
- If you used the account context menu, the loan you want to modify may be displayed. If not, use the search dialog that appears to search for and select the loan you want to modify.

Copyright © 2024, Oracle and/or its affiliates.

- In the **Renegotiate Future Payments** section, change the interest rate (if allowed), the number of payments, or the payment amount. Note that either the number of payments or payment amount must be blank.
- Click **Calculate**. The **Forecasted Schedule** is recalculated along with either the new payment amount or the new number of payments (depending on which was left blank).
- If the renegotiated payments are satisfactory to the customer, save the changes.

When you save the changes, appropriate entries are made in the **Override Interest Rate** and **Payment Amount** grids.

Note: Saving Loan Modifications. You must save the renegotiated payments after clicking Calculate for the loan modifications to be effective.

How To Adjust Loans

You issue ad hoc adjustments if you need to change a loan's principal and/or current balance outside of the normal billing and payment functions.

An adjustment can:

- Change a loan's payoff balance.
- Change a loan's current balance (i.e., how much the customer thinks they currently owe).
- Change both the loan's payoff and current balance.

Fastpath: For details about the adjustment types to apply, refer to Adjusting Loan Amounts. For information about creating adjustments, refer to Adjustments.

How To Stop A Loan Contract

Normally, a loan contract is automatically stopped when the bill segment for the last payment in the amortization schedule is generated for the loan (refer to Finalizing And Closing A Loan).

If a customer decides to payoff a loan before the term is complete, you can:

- Transfer to the ontract maintenance page for the loan contract, specify an **End Date** for the loan, and click Save. This transitions the loan contract to `pending stop`.

Note: End Date. The loan contract cannot be stopped until the specified end date is reached, so do not specify an end date that is in the future unless you do not want to stop the loan until that date.

Once the loan is in pending stop and the end date has been reached, the system transitions the loan contract to `stopped`. On the contract's normal bill cycle, the system final bills the contract for any outstanding balance plus accrued interest charges.

If you want to stop the loan and create the final bill immediately, complete these additional steps:

- On the loan's contract maintenance page, click the Stop Contract button to transition the loan contract to `stopped`.
- Manually, create a bill for the loan contract. Remember that if you only want to generate a bill segment for a specific contract (e.g., the loan contract), transfer to the bill segment page. From there, you can generate and freeze a bill segment for a specific contract. When finished, return to the bill page and complete the bill.

When the customer pays the final bill, the loan contract is closed by the system.

Fastpath: For more information about contract states, refer to The Lifecycle Of A Contract.

Copyright © 2024, Oracle and/or its affiliates.

Chapter 13

Deposits

Topics:

- The Big Picture Of Deposits
- Deposit Review

In this section, we describe how to manage your deposits.

The Big Picture Of Deposits

The topics in this section provide background information about a variety of deposit issues.

Fastpath: We strongly recommend familiarizing yourself with the topics described in The Financial Big Picture to fully appreciate how deposits fit into the system's financial architecture.

Cash Deposits

Before you can bill a customer for a deposit, there must be a deposit contract. Deposit contracts will be created by a CSR using Start/Stop Service (the same page used to create all other types of contracts).

Deposit contracts behave just like any other contract in that:

- A deposit contract must reference a contract type. Refer to Contract Type Controls Everything for information about how contract type controls a contract's behavior.
- Bill segments will be produced to bill the customer for a deposit. Note, bill segments associated with deposit contracts will appear on the same bill as other bill segments related to its account (i.e., a bill could contain a combination of utility, deposit, and non-utility bill segments).
- When a customer makes a payment, it will be distributed amongst its account's contracts based on each contract's payment distribution priority. This means a single payment could relieve receivable accounts (associated with utility contracts) and increase a payable account (associated with the deposit contract). Refer to Distributing A Payment Amongst An Account's Contracts for more information.
- You can use a transfer adjustment to relieve debt on some other type of contract (by transferring all or part of the deposit's credit balance to another contract).
- You can use an A/P adjustment to cause a check to be created if you need to manually refund a deposit.
- When you no longer require the deposit, a CSR will stop the deposit contract using Start/Stop Service (the same page used to stop all other types of contracts). Refer to Refunding Deposits for information about how the refund actually takes place.

Fastpath: Refer to Deposit Class Controls Everything **for information** about deposit interest, the recommended deposit amount algorithm, and how deposits are refunded.

Note: An account can have many deposits. You can view all deposits (both cash and non-cash) linked to an account on Account - Deposits.

Refunding Deposits

To refund a deposit, you simply need to change the deposit contract's state to `pending stop`. The system does everything else (i.e., calculates interest and refunds the deposit to the customer).

There are three ways in which a deposit contract's state can become `pending stop`:

- The Deposit Refundbackground process (referred to by the batch control ID `DEPRFND`) make a deposit contract `pending stop` when it detects that the customer meets the automatic refund criteria. Refer to Defines The Conditions That Cause The System To Automatically Refund A Deposit for a description of how to define the conditions that control automatic refund.
- An operator, at their discretion, may refund a deposit by stopping the deposit contract using Start/Stop Service (the same page used to stop all other types of contracts). This page simply changes the deposit contract's state to `pending stop`.
- The system changes the state of a deposit contract to `pending stop` when it recognizes that all other contract's of its deposit class have also been `stopped`. This may be necessary, for example, when a customer's utility contracts

Copyright © 2024, Oracle and/or its affiliates.

have been severed for non-payment. Refer to Deposit Seizures When Other Contracts Are Stopped for more information.

The system changes the state of a `pending stop` contract to `stopped` when the process responsible for Finalizing Pending Stops next runs.

When a deposit contract becomes `stopped`, the system calculates interest through the stop date. Refer to Controls Interest Calculation for a description of how interest is calculated.

Note: If you can't wait for this process to run, simply display the deposit contract on Contract- Main Information and press the stop button.

When the `stopped` deposit contract is final billed, the system produces a final bill segment that shows the deposit amount being returned to the customer. When the bill segment's bill is completed, the deposit is refunded to the customer as per the deposit refund algorithm on the deposit's deposit class. Refer to Controls How Deposits Are Refunded To A Customer for a description of how a deposit is refunded.

CAUTION: None of the base package deposit refund algorithms create an A/P adjustment when the deposit contract is final billed. Rather, they depend on the write-off preprocessing to actually refund the deposit to the customer.

Partial Refunds

You can perform a partial refund by creating an adjustment for the deposit contract. If you want to refund the adjustment with a check, this adjustment should be an A/P adjustment (i.e., one that is interfaced to A/P). If you want to refund the adjustment by offsetting some other contract's debt, create a transfer adjustment (transferring from the deposit contract to the contract to be relieved).

After creating the adjustment, you'll find that the deposit contract has a positive current amount due (because you debited the deposit contract to refund the credit amount). You should create another adjustment to return the current amount of the deposit to zero. It's important to use an adjustment type that only affects current amount when you do this.

After getting the deposit's current amount back to an expected amount, you will have to change the deposit contract's Total Amount To Bill to reflect the amount of deposit you want to hold for the customer. So, for example, if you were holding a $500 deposit and you want to refund $100, you should change the deposit contract's Total Amount To Bill to be $400.

Total Amount To Bill

Most deposit contracts contain a "Total Amount To Bill" field. The label for this field is defined on the contract type (on the Billing tab). For deposit contracts, it should be labeled something liked **Cash Deposit Amount**. This field should reflect the deposit amount you WANT to hold on this contract. Please be aware of the following in respect of this field:

- If the payoff balance (i.e., the current deposit held) is less than the Total Amount To Bill, the system will generate a bill segment to charge the customer for the deposit.
  - If the deposit contract uses a recurring charge (i.e., you bill the total amount in installments), the amount billed to the customer will not exceed the recurring charge amount (unless the bill segment is prorated).
  - If the deposit contract does not use a recurring charge, the amount billed will equal the difference between Total Amount To Bill and the contract's payoff balance.
- If you need to bill an incremental deposit on an existing deposit contract, simply change the deposit contract's Total Amount To Bill.
- If the payoff balance (i.e., the current deposit held) is greater than the Total Amount To Bill, the system will NOT refund the excess deposit.

Note: Contract Type and Total Amount to Bill. A deposit contract's contract type controls if the deposit contract uses Total Amount to Bill. Typically, the only contract types that indicate this field can be used are loan and deposit contracts.

Copyright © 2024, Oracle and/or its affiliates.

Current Balance versus Payoff Balance

CAUTION: If you do not understand the difference between payoff balance and current balance, refer to Current Amount versus Payoff Amount.

The current balance on a deposit contract contains the amount of the deposit that has been billed, but not paid. The payoff balance on a deposit contract contains the amount of deposit being held. The payoff balance is a negative number because your company owes this money to the customer.

The financial ramifications of a deposit contract are predictable (if you're an accountant).

The following table indicates the payoff and current balance when the financial transactions are created for the respective events:

Event	GL Accounting	Arrearage Rule	Effect On Payoff Amt	Effect On Current Amt	Payoff Balance	Current Balance
Deposit billed	N/A - the GL is not affected when a deposit is billed	$100 starts aging	0	+100	0	100
Payment received	Cash 100 Deposit Payable <100>	$100 relieved accordingly	-100	-100	-100	0
Interest calculated	Interest Exp 5 Deposit Payable <5>	N/A	-5	0	-105	0

Note: It's important to be aware that everything that is shown in this table is controlled by how you set up the deposit contract type's bill segment type, payment segment type, and adjustment types. Refer to Contract Type Controls Everything for how to do this.

The following points describe the events in the above table:

- **Deposit billed.** In this example, the customer is billed for $100 deposit.
  - The customer really thinks they owe the billed amount, $100. Therefore, current amount is affected. However, if the customer was to cash out, they wouldn't owe your organization anything, therefore payoff amount is not affected.
  - Notice that the GL is not affected when the deposit is billed. This is because most organizations do not show a receivable for billed deposits (as it's not a true receivable).
  - Because current amount changed by $100, arrearage history is affected accordingly.
- **Payment received.** With any luck, the client will pay the $100 that was billed.
  - The payment has a normal affect on the GL (debit cash, credit deposit payable). The deposit payable GL account is defined as the deposit contract type's Distribution Code.
  - The amount the customer thinks they owe decreases by $100, therefore current amount is affected by the payment amount. And, if the customer was to cash out, your organization would owe the customer $100, therefore payoff amount is affected by the payment amount.
  - Because current amount changed by $100, arrearage history is affected accordingly.
- **Interest calculated.** In this example, the system calculates interest of $5.
  - The interest is posted to the GL (the interest expense distribution code is defined on the respective adjustment type).

Copyright © 2024, Oracle and/or its affiliates.

- The interest amount didn't affect how much the customer thinks is due them. Therefore current amount is unaffected. However, if the customer was to cash out, your organization would owe them $100 + $5 (the interest) therefore payoff amount is affected by $5.
- Because current amount is not changed, arrearage history is not affected.

Multiple Deposits Linked To A Single Account

The system has been designed to allow your customers to have multiple deposits. This is necessary when a deposit is restricted to a specific type of debt. For example, if separate deposits are held for regulated and unregulated debt (and a customer could hold a combination of regulated and unregulated debt), you'd need one deposit for regulated debt and another for unregulated debt.

Note: Deposit class. A deposit's deposit class controls the contracts that are covered by a given deposit. A deposit's deposit class is defined on its contract type. Refer to Designing Your Deposit Classes for more information.

It's important to be aware that if your company has multiple classes of deposits, the system will recommend and refund deposits for each individual deposit class. For example, if you have both regulated and unregulated deposit classes, the system will recommend separate deposits for each class. And at refund time, a deposit will be distributed to only those contracts associated with its deposit class (i.e., a regulated deposit will not be used to satisfy unregulated debt).

Deposits and Credit & Collections

The account debt monitor monitors deposit contracts just as it monitors every other contract for overdue debt. If the amount of debt on the deposit contract violates your collection criteria, a collection process will start.

Fastpath: Refer to The Lifecycle Of A Collection Process And Its Events **for more information**.

It's important to be aware that deposit debt can be treated differently from other types of debt linked to an account by creating a specific debt class for the deposit contract type.

Fastpath: Refer to Designing Your Collection Procedures for more information about how to have different collection criteria for different debt classes.

The collection process associated with a deposit contract is probably rather simple - you probably will want a ToDo entry generated to advise an operator that a customer hasn't paid their deposit.

Please see Deposit Seizures When Other Contracts Are Stopped for a description of how the system seizes an active deposit contract when normal (i.e., non-cash deposit) contracts are stopped.

After a deposit contract is `stopped`, it will be "final billed" the next time the account is billed. Refer to Refunding Deposits for more information about how deposit refunds are "final billed". If a credit balance remains on the deposit contract after it is final billed, the Write-Off Monitor will process the deposit contract just like it processes all unpaid and final billed contracts. If you set up the deposit contract's write-off controls properly (i.e., you plug in the appropriate refund and write-down algorithms), the write-off monitor will refund / write-down the credit balance (thus causing the deposit contract to close).

Deposit Seizures When Other Contracts Are Stopped

Consider a situation when a service provided to a customer is stopped due to non-payment. At some point, if the customer doesn't pay, you'll want to seize the deposit and apply it to the overdue debt. To implement this, the system simply stops the deposit contract when it stops the last contract in its deposit class (refer to Finalizing Pending Stops for more information about stopping contracts). Then, when the account is next billed, the standard deposit refund process will offset outstanding debt before it is refunded to a customer.

Copyright © 2024, Oracle and/or its affiliates.

Refer to Refunding Deposits for a complete description of how the system refunds a stopped deposit contract's balance to other contracts in its deposit class.

Note: Deposit class refund method is the key. It's important to stress that the only way the system will use a deposit to offset overdue debt is if the deposit's deposit class has a refund method that does this. If you've set up a deposit class' refund method to cut a check for the entire deposit amount, the system will not offset outstanding debt. Refer to Deposit Class - Refund Method for more information.

Non-Cash Deposits

When an account is required to post a deposit, they can remit cash or some other type of surety (e.g., letters of credit, surety bonds, 3 rd party deposits). When cash is remitted, a deposit contract is used. When some other type of surety is used, a *non-cash deposit* must be created.

Non-cash deposits are held in respect of an account and an account may have an unlimited number of non-cash deposits (note: cash deposits are held in respect of a deposit contract that is linked to remitting account). Refer to Account - Deposits for more information.

Each non-cash deposit must reference a non-cash deposit type. Besides defining the type of surety, the non-cash deposit type also controls:

- Whether such types of non-cash deposits can be highlighted when they are due to expire via the creation of a ToDo entry (refer to TD-NCDEX for more information about this ToDo entry).
- Whether a reference to a cash-deposit (i.e., contract) must be specified. Refer to 3rd Party Deposits for more information about how a cash deposit can be referenced by another party.

Fastpath: Refer to Setting Up Non-Cash Deposit Types for more information.

In addition to non-cash deposit type, each non-cash deposit must reference a deposit class. Why? Because the system amalgamates cash and non-cash deposits when it determines if an account is holding an adequate deposit.

Fastpath: Refer to What Do Deposit Classes Do? for more information.

If you need to use a non-cash deposit to satisfy an account's debt, you must "cash in" the non-cash deposit. When the cash arrives, create a payment and apply it to the customer's outstanding contracts.

3 rd Party Deposits

A 3 rd party deposit arises when a 3 rd party remits cash to cover the deposit needs of one or more accounts. Both cash and non-cash deposits must be created to record a 3 rd party deposit. The following example will explain how to do this.

Assume the Salvation Army remits $5,000 on behalf of 50 customers (where each customer's account is allocated $100). In this situation, you'd create the following information in the system:

- The Salvation Army must have an account with a deposit contract.
- When the Salvation Army remits the funds:

CAUTION: If you do not understand the difference between payoff balance and current balance, refer to Current Amount versus Payoff Amount.

- Create an adjustment to "bill" the deposit contract (causing the deposit contract's current balance to be $5,000 and the payoff balance to be 0).
- Add a payment for the $5,000 against the Salvation Army's account. This payment will cause the deposit contract's current balance to be 0 and the payoff balance to be -$5,000. Note, if you don't mind the Salvation Army's current balance to be -$5,000 after the payment is made, you wouldn't have to create the adjustment to "bill" the deposit.
- Create a non-cash deposit for each of the 50 accounts being covered by the cash deposit. On each non-cash deposit, define the appropriate non-cash deposit type (e.g., 3 rd party deposit) and amount $100 each.

Copyright © 2024, Oracle and/or its affiliates.

- Interest will be applied to the Salvation Army's deposit contract as per the interest algorithm on the deposit contract's deposit class.
- If you need to use the Salvation Army's payment to payoff overdue debt, you will use a transfer adjustment(s) to transfer from the deposit contract to the respective overdue contract(s).

Note: Important! Be aware that the system will allow the sum of 3 rd party deposits to exceed the amount of the cash deposit (in our previous example, the system would allow you to create $6,000 worth of 3 rd party non-cash deposits even though only $5,000 of cash was remitted).

Deposit Class Controls Everything

A deposit contract's contract type references a deposit class. The topics in this section describe how deposit class controls the behavior of a deposit contract.

Fastpath: Refer to Setting Up Deposit Classes for a complete description of the business rules governed by a deposit class.

Controls Interest Calculation

Interest is applied to a deposit contract when:

- A deposit contract is `stopped`. Refer to Refunding Deposits for a description of how a deposit contract can be stopped.
- The `Apply Interest` background process (referred to by the batch control ID `DEPINTRF`) determined that sufficient time has passed since interest was last calculated. The amount of time between interest calculations is defined on the deposit class.

A deposit contract's deposit class controls the following interest calculation functions:

- The frequency of interest calculation.
- The algorithm used to calculate the amount of interest (note, the interest rate is defined using a bill factor plugged in on this algorithm).
- The method used to refund the interest.

Note: Adjustments are used to apply interest. Interest is applied to a deposit contract using an adjustment (the adjustment type is also defined on the deposit class).

Controls The Recommended Deposit Amount

The following events cause the calculation of the recommended deposit amount:

- The `Deposit Review` background process (referred to by the batch control ID `DEPRVW`) compares an account's existing deposit (if any) to the recommended amount.
- A user requests a deposit calculation for a new deposit contract by pressing the button on the Start Confirmation.

A deposit contract's deposit class controls the algorithm used to derive the suggested deposit amount.

Defines The Conditions That Cause The System To Recommend A New or Additional Deposit

The Deposit Review background process (referred to by the batch control ID `DEPRVW`) will recommend an additional deposit be billed if the customer doesn't satisfy the definition of a *good customer*.

It's important to be aware that if your company has multiple classes of deposits, the system will recommend deposits for each individual deposit class used by an account. For example, if an account has both regulated and unregulated deposit classes, the system will recommend separate deposits for each class.

When the Deposit Review process analyzes an account's debt, it uses the following deposit class attributes:

Copyright © 2024, Oracle and/or its affiliates.

- The definition of a good customer (the system only recommends deposits for bad customers).
- The tolerance percent that must be exceeded before the system will recommend an additional deposit for a customer. This tolerance percent prevents the recommendation of small amounts.
- The method used to derive the recommended deposit amount.

Assuming the C1-CR-UP-DRR review method algorithm is specified on the deposit class, the system does not automatically produce bills when an additional deposit is needed. Rather, the system's recommendations appear on the Deposit Review page. To implement a recommendation, an operator should change the deposit SA's Total Amount To Bill or add a new deposit SA.

Note: Non-cash deposits are included. It's important to be aware that the system compares the recommended deposit amount against the sum of deposits on hand for a given deposit class. When amalgamating the total deposit on hand, the system includes both cash (i.e., deposit SA's payoff balance) and non-cash deposits.

Defines Automatic Refund Conditions

The Deposit Refundbackground process (referred to by the batch control ID `DEPRFND`) will refund a deposit to a customer when conditions defined on the deposit SA's deposit class are met. For example, a deposit's deposit class may indicate the system should automatically refund a deposit after the deposit has been held for 6 months and the customer is a good customer (the definition of a "good customer" is also on the deposit class).

Controls How Deposits Are Refunded To A Customer

The method used to refund a deposit to a customer is defined on the deposit SA's deposit class. For example, a deposit's deposit class may indicate the system will first apply the deposit to outstanding debt first and, if funds remain, cut a check.

It's important to be aware that if your company refunds deposits by first offsetting outstanding debt, the system will only offset debt within an individual deposit class. For example, if a customer has two deposits - one for regulated debt, the other for unregulated debt - the regulated deposit will only be used to offset regulated debt.

Deposit Background Processes

The topics in this section describe the background processes that automate deposit processing.

Deposit Interest

The Deposit Interest background process (referred to by the batch control ID `DEPINTRF`) examines all cash deposit contracts (i.e., contracts with a contract type with a special role of `Cash Deposit`). If enough time has passed since interest was last calculated, interest will be calculated using the contract's deposit class' Interest Refund Algorithm. Note: interest will be automatically calculated every X months where X is defined in the deposit class' Months Between Interest Refund.

Deposit Refund

The Deposit Refund background process (referred to by the batch control ID `DEPRFND`) examines all cash deposit contracts (i.e., contracts with a contract type with a special role of `Cash Deposit`). If the account meets the contract's deposit class' Refund Criteria Algorithm, the system changes the contract's state to `Pending Stop`. Refer to Refunding Deposits for a description of what happens next.

Review Deposits

The Review Deposits background process (referred to by the batch control ID `DEPRVW`) examines all accounts with contracts that are governed by a deposit class (i.e., contracts with an contract type that references a deposit class). If the account fails the deposit class' Good Customer Algorithm, the system calculates the recommended deposit amount (using the deposit class' Deposit Recommendation Algorithm). If the recommended amount exceeds the amount of deposit currently requested, the system will request an additional deposit. The deposit class' **Review Method Algorithm**

Copyright © 2024, Oracle and/or its affiliates.

is used to determine what action to take if the system requests an additional deposit. The system's recommendations appear on the Deposit Review page.

You may optionally provide a Deposit Class as input to restrict the review to account whose contracts reference that deposit class.

Note: Preventing small recommendations. The system uses the deposit class's Review Tolerance Percentage to prevent the recommendation of small deposits by the Deposit Review background process. For example, if this field contains 10(%), the system would only recommend an additional deposit if the existing requested deposit amount is less than 90% of the recommended amount.

Note: Multiple deposits could be recommended for an account. Because an account can have contracts that belong to multiple deposit classes, it is possible for a multiple recommendations to be generated for an account.

Deposit Review

The Deposit Review page shows all accounts for which an additional deposit is recommended.

Note: The Review Deposits background process inserts rows on this query.

Note: Multiple deposits could be recommended for an account. Because an account can have contracts that belong to multiple deposit classes, it is possible for a multiple recommendations to be generated for an account.

Open this page using **Financial Query, Deposit Review**.

Description of Page

Use **Account ID** and / or **Deposit Class** to filter the rows. Each row displays and account / deposit class that holds an inadequate deposit.

Note: Rows are also displayed for errors. Please be aware that if the Recommend Additional Deposit background process encounters an error when it attempts to calculate an account's recommended deposit, these errors also appear as rows in this query.

Press the button to add a new deposit contract for the customer. Pressing this button causes Start/Stop Service to open. When this page opens, you should create a new deposit contract for the customer (by referencing the appropriate division / contract Type).

If the customer has one or more deposit contracts, the drill button will be enabled. Pressing it will cause the Contract - Main Information page to open. When this page opens, you can change Total Amount To Bill to reflect the total amount of deposit you want to hold. When billing next runs, it will bill the customer for the marginal difference between the current deposit held and the deposit amount to bill.

Fastpath: Refer to Defines The Conditions That Cause The System To Recommend A New or Additional Deposit for how rows are added to this query.

Copyright © 2024, Oracle and/or its affiliates.

Chapter 14

Case Management

Topics:

- Background Topics
- Maintaining Cases

Note: Separate module. Please note that the Case Management functionality is associated with a separate `Case Management` module. If this module is not applicable to your business you may turn it off. Refer to Turn Off A Function Module for more information..

Your organization can use case management functionality to manage a variety of situations, such as:

- a high-bill complaint,
- a bankruptcy,
- a customer's request for literature,
- a contractor's request to extend a line,
- a customer's rejection of a quote,
- a customer's request to change information on a future date,

Note: Your implementation controls how your cases behave. Since virtually all aspects of case management functionality are controlled by your implementation, you can use it to handle a myriad of business requirements. Refer to Defining Case Management Options for the details.

Background Topics

The topics in this section provide background information about case management functionality.

Case Type

A case is controlled by the case type that it references. Some aspects of the case that are controlled by the case type include:

- The lifecycle of the case
- The applicability of a person and account
- Which fields are required during the various stages of a case's life
- Which users can access cases of a given type
- Additional processing that should take place when a case enters or leaves a given state

Refer to Case Type Controls Everything for a description of the business rules that you control when you set up your case types.

Creating Cases

To create a case, open the case management page in add mode, select the appropriate case type, fill in the other necessary fields, and click save. Oracle Revenue Management and Billing also includes scripts that can help create cases. Refer to Scripts and Cases for more information.

You can also develop plug-ins and user exits so that the system will create cases when specific events take place.

Please refer to Maintaining Cases for a detailed description of the case page and field descriptions.

Log Information

A case log contains an entry for every recorded event during the lifecycle of the case. There are two types of log entries:

- **Automatic entries.** The system automatically creates an entry in the log when a case is created or there is a status change. Users cannot modify or delete these log entries.
- **Manual entries.** Users can add manual entries to record significant events at their discretion.

Refer to Case - Main for the page used to view the log.

Alerts

The alert zone in the dashboard indicates whether the person / account in context have cases in a state that is configured as "alertable" if you plug-in the appropriate algorithm on the installation record. Refer to Alert Info Is Controlled By An Installation Plug-In for the details.

To Dos and Cases

You can configure the system to inform users of cases that require their attention. Refer to ToDo's and Cases for the details.

Copyright © 2024, Oracle and/or its affiliates.

Reports

Several reports summarize case information.

Maintaining Cases

Your organization can use the case management functionality to manage a variety of situations, such as:

- a high-bill complaint,
- a bankruptcy,
- a customer's request for literature,
- a contractor's request to extend a line,
- a customer's rejection of a quote,
- a customer's request to change information on a future date,

Note: Since an implementation can configure virtually every aspect of a case, the look and feel of the case management page are dynamic based on how case types have been configured.

Case - Main

Open the case management page by selecting **Customer Management, Case** from the main menu.

Main Information

Case Info contains a concatenation of important information about the case. **Case ID** is the system-assigned unique identifier of the case. These values only appear after the case is added to the database.

Note: Formatting may be performed by a plug-in. Refer to Case Info May Be Formatted By An Algorithm for more information.

Case Type defines the type of case. This field is protected after the case has been added. Refer to Case Type Controls Everything for more information.

Status shows the current state of the case. Refer to Case Lifecycle for more information.

If the case has any ToDo entries associated with it, a hyperlink will appear summarizing the number and status of the entries. You can select the link to view these entries. Refer to ToDo's and Cases for more information.

Date / Time Opened shows when the case was created. If the case is in a final state, the **Date / Time Closed** will also be shown. Refer to One Initial State and Multiple Final States for more information.

Script references the business process assistant script that can help you work on the case. You can click on the hyperlink to execute this script. This field is hidden if no script is associated with this case's current Status. Refer to Scripts and Cases for more information.

The **Actions** section contains buttons you can use to action the case. This area is suppressed if no actions can be performed for the case given its current status. A button will be gray if you do not have security rights to the action. Refer to Buttons Are Used To Transition A Case for more information.

When the user adds a new case or changes the state of a case manually the system attempts to auto-transition the case to subsequent statuses as necessary. If auto-transition rules apply to the new state (and to subsequent ones) they would be executed right away. If the case has been automatically transitioned to its current state the following message area appears right after the action area. If an error is encountered in the process a corresponding error message is displayed in the Auto-transition Message Area. Refer to Automatic Transition Rules for more information.

Copyright © 2024, Oracle and/or its affiliates.

> **Note:** The Auto-transition Message Area only appears when a case has just been automatically transitioned to its current state online. It only remains visible until the data for the case is refreshed.

Use the **Comments** area to describe anything interesting or unusual about the case.

Person references the person associated with this case. This field is suppressed if the Case Type indicates that a person is not allowed on cases of this type. This field is protected if the case is closed (i.e., if its status is defined as being final). This value is defaulted to the Person currently displayed in the Dashboard.

Account references the account associated with this case. This field is suppressed if the Case Type indicates that an account is not allowed on cases of this type. This field is protected if the case is closed (i.e., if its status is defined as being final). This value is defaulted to the Account currently displayed in the Dashboard.

Responsible User references the user who has overall responsibility for the case. This field is suppressed if the Case Type indicates that a responsible user is not allowed on cases of this type. This field is protected if the case is closed (i.e., if its status is defined as being final).

The **Contact Information** area is used to define how to contact the individual who initiated the case. These fields are suppressed if contact information is not allowed on cases of this type. The fields cannot be modified if the case is in a closed status. The following information can be defined.

- **Contact Person** references the person who should be contacted. This defaults to the Person defined above.
- **Preferred Contact Method** indicates how the person prefers to be contacted. The following options are available:
  - `Email.` If this method is selected, the Contact Person's email address appears adjacent.
  - `Fax.` If this method is selected, the Contact Person's fax number appears (the system knows which of a person's phone numbers is a fax number by the phone type).
  - `Not Applicable.` If the person doesn't want to be contacted (or the contact method is unusual), use this option. You might want to consider describing the situation in the **Contact Instructions**.
  - `Phone.` If this method is selected, the Contact Person's phone number appears.
  - `Postal.` If this method is selected, the Contact Person's address appears. Refer to Where Does The Address Come From for a description of which address the system displays.
  - `SMS.` If this method is selected, the Contact Person's short message service (SMS) information displays.

`Contact Instructions` indicates any special instructions indicated for when or how to return a call.

`Callback Phone Type`, `Callback Phone Number`, and `Extension` indicate the actual phone number and phone type that should be used to call.

Characteristics

The **Characteristics** grid contains fields that contain important information about the case. This grid is suppressed if the Case Type does not use characteristics. This grid is protected if the case is closed (i.e., if its status is defined as being final).

You can only choose **Characteristic Types** defined as permissible for this case type. Refer to *Additional Information* for more information. If you attempt to use an **Action** to transition a case into a status that requires additional information, a pop-up will prompt you for the additional fields. Refer to Required Fields Before A Case Enters A State for more information.

Case - Case Portal

To view additional information associated with a case, open by selecting **Customer Management, Case** from the main menu and then the **Case Portal** tab.

General Information

Case ID is the system-assigned unique identifier of the case.

Case Info contains a concatenation of important information about the case.

Copyright © 2024, Oracle and/or its affiliates.

Object Portal

When information associated with the case has been stored in the case Character Large Object (**CLOB**) field (such as the data associated with a market message), the **Object Portal** displays this information as-is. You cannot modify the case CLOB field data from this page, but case type algorithms can be configured to work with this data. For information on mapping data into CLOB fields, refer to the Business Object tips document **Schema Nodes and Attributes**.

Case - Log

To view the log entries associated with a case, open by selecting **Customer Management, Case** from the main menu and then the **Log** tab.

Description of Page

The **Log** grid displays log entries (in reverse chronological order) that audit the progress of the case. Please note the following about the entries in this grid:

- The system automatically creates log entries when a case is created, every time its status changes, or whenever an error occurs during the transition of a case. Algorithms plugged into a case type state can also be configured to create log entries. You cannot modify or delete the log entries.
- You can manually add a log entry by pressing the + button and enter the **Details**. You cannot modify or delete this information after saving it.
- The following information appears in the grid:
  - **Log Date/Time** contains the date and time the log entry was created.
  - **Details** contains user-specified or system-generated information about the log entry.
  - **Related Object** is populated on log entries that were created to record the creation of some other object. For example, if a case plug-in creates a customer contact, the related object contains information about the customer contact. Please note that if the object description is shown in blue, you can click on the object's description to drill down to the object.
  - **Log User** contains the user who caused the log entry to be created.
  - **Log Type** indicates how the log entry was created. The possible values are:
    - `Created.` Only one entry per case will be of this type. This entry is automatically created by the system when the case is created.
    - `User Details.` A user creates this type of entry manually.
    - `Status Transition.` This type of entry is created by the system each time the case's status is changed (as a result of pressing one of the Action buttons).
    - `System`. An algorithm on the case type state may be designed/configured to create this type of entry when it does something auditable. For example, an algorithm that creates a customer contact could insert a log entry to show that it did this (it will also populate the FK to the customer contact, which will appear in the **Related Object** column).
    - `Transition Error`. The background status transition process creates an entry of this type if an error occurs when automatically updating a case to a subsequent status (as directed by either an Enter Status algorithm or an Auto-Transition algorithm). An entry of this type lists the case states involved with the failed transition, and the algorithm in which the error was generated.
    - `Exception`. The background status transition process creates an entry of this type under the same circumstances as a Transition Error. A log entry of this type indicates why the background transition failed.

Note: **Note**. The values for this field are customizable using the Lookup table. This field name is CASE_LOG_TYPE_FLG.

Copyright © 2024, Oracle and/or its affiliates.

Chapter 15

Job Streams

Topics:

- Job Stream Summary
- Job Stream Details

The term "job stream" refers to a suite of batch jobs that run periodically. Most organizations have multiple job streams. For example,

- You'll have a job stream that contains batch jobs that run nightly
- You'll have a different job stream that contains the hourly batch jobs
- You'll have a different job stream that contains the batch jobs that extract data for your data warehouse
- Etc.

The topics in this section describe how to execute and monitor your job streams.

Fastpath: Refer to The Big Picture of Scheduling Batch Jobs for a description of how to set up the system to enable this functionality.

Note: Only used by some organizations. Many organizations use a third party tool to manage the periodic execution of batch jobs rather than the system batch scheduler. If this is true for your organization, this section is not applicable.

Note: Separate module. Please note that the batch scheduler functionality is associated with a separate `Workflow Scheduling` module. If this module is not applicable to your business, you may turn it off. Refer to Turn Off A Function Module for more information..

Job Stream Summary

This page presents a summary of your job streams. Open this page using **Batch Scheduler, Job Stream Summary**.

Description of Page

This page summarizes the status of your job streams.

Note: Only job streams defined on the Feature Configuration for the Batch Scheduler appear on this page.

The following information is displayed for each job stream:

- **Job Stream.** This is the description of the job stream. Click the hyperlink to transfer to the job stream's workflow process template (the workflow process template defines the number and type of batch jobs in the job stream as well as their dependencies).
- **Cancel Button.** If the job stream is executing, a **Cancel** button appears. Pushing this button will cancel the execution of any batch jobs in the job stream *that have not already started*. Batch jobs that are currently executing will continue to execute until they conclude or are stopped in the operating system.
- **Start Button.** If the job stream is not executing, a **Start** button appears. Pushing this button submits the request to execute the job stream's batch jobs. The job stream's batch jobs will only execute when the batch submitter next runs.
- **Last Completed Run.** This is the start date / time when the job stream last executed. In addition, the job stream's total elapsed time is displayed. Click the hyperlink to transfer to the Job Stream Details page where you can see the execution status of each batch job in the job stream.
- **Run In Progress.** This column only contains information if a job stream is executing. The date / time that the job stream was requested is shown as well as a colorful status bar. Clicking the hyperlink under the date / time transfers you to the Job Stream Details page where you can see the execution status of the job stream's batch jobs.
- The status bar provides a visual clue as to the execution status of the job stream's batch jobs. The bar's segments and their colors are used to highlight the relative number of batch jobs in a given state. For example, it will highlight the relative number of `Pending` versus `Complete` versus `Error` batch jobs.

You can hover the mouse over a section of the bar to see the number of batch jobs in each state. You can click on a segment of the bar to drill down to the Job Stream Details page. When this page appears, only the batch jobs in the respective state are displayed.

The message `Other(s) running` appears if more than one instance of this job stream is running (this should be rare).

- **Next Scheduled Run.** This is the start date / time when the job stream is next scheduled to run. Click the hyperlink to transfer to the Job Stream Creation Schedule page where you can define when the job stream runs. You can also use this transaction to hold prescheduled runs.

Job Stream Details

Job streams execute many times over their lifetime. For example, the nightly job stream executes every night; the hourly job stream executes hourly; the data warehouse extract executes weekly; etc.

Every time a job stream executes, a workflow process is created. The workflow process's events cause the job stream's batch jobs to be submitted.

This page shows the status of the job stream's workflow events and the batch jobs that they submit. Open this page using **Batch Scheduler, Job Stream Details**.

Description of Page

Job Stream is a concatenation of important information about the execution of the job stream (e.g., its description, status, start date).

Job Stream ID is the system-assigned unique identifier of an execution of the job stream.

Copyright © 2024, Oracle and/or its affiliates.

> **Note:** The Job Stream ID is the unique identifier of the workflow process used to manage the execution of the job stream's batch jobs.

The colorful status bar summarizes the execution status of the job stream's batch jobs. This bar's segments and their colors are used to highlight the relative number of batch jobs in a given state. For example, it can highlight the relative number of `Pending` versus `Complete` batch jobs (`Pending` jobs are one color, `Complete` jobs another).

You can hover the mouse over a section of the bar to see the number of batch jobs in each state. You can click on a segment of the bar to cause the grid to be refreshed with batch jobs in the respective state. For example, if you click on the color used to highlight jobs in `Error`, only these jobs will appear in the grid. You can achieve the same results by selecting the desired **Event Status**.

If the job stream is executing, a **Cancel Job Stream** button appears. Pushing this button will cancel the execution of any batch jobs in the stream *that have not already started*. Batch jobs that are currently executing will continue to execute until they conclude or are stopped in the operating system.

The grid contains a summary of the job stream's workflow events and the batch jobs that they submit. The following information is displayed for each workflow event:

- **Event** describes the workflow event that submits the batch job (the code following the description is the workflow event's type).
- The colorful box that prefixes the event provides a visual clue of the event's status (e.g., `Pending` events are one color, `Waiting` another, etc.). These colors correspond with the "status bar" at the top of the page.

Click on the hyperlink to transfer to the workflow event. You might need to display a workflow event if:

- You need to cancel a `Pending` or `Waiting` event. To do this, display the event and change its status to `Canceled`. Note, if the `Waiting` event is associated with a batch job that is currently executing, canceling the event will NOT abort the execution of the batch job. The only way to abort the execution of a batch job that is executing is via the operating system.
- You need to restart a batch job that's in `Error`. To do this, display the workflow event and change its status to `Pending`.
- **Batch Run** provides an overview of the batch job submitted by the workflow event:
- • If the corresponding workflow event is `Pending`, the batch job has not been submitted yet. There are two reasons why an event may be `Pending`:
- If the event is waiting for the batch submitter to run, the message `Starting soon...` appears. The next time the batch submitter runs, the batch run will be executed.
- If the event is waiting for the successful completion of other events, a message describing the events on which it is waiting is shown. You define the dependencies between events (i.e., batch jobs) when you set up the job stream's workflow process template.
  - If the corresponding event is `Waiting`, the batch job is currently executing and a summary of the execution status is shown. You can click on the hyperlink to transfer to the Batch Run Tree where the details of the run's parallel threads and errors are shown. Note, a summary of the various threads and errors is shown in the next column.
  - If the corresponding event is `Complete`, `Failed`, or `Error`; the batch job has completed and a summary of its execution is shown. You can click on the hyperlink to transfer to the Batch Run Tree where the details of the run's parallel threads and errors are shown.
  - **Batch Run Details** provides more extensive details about a batch run's execution. The following information is shown:
  - • You can configure the workflow event to execute multiple parallel threads of a given batch job. For example, rather than running a single instance of the BILLING batch job, you can indicate 40 parallel threads should run (thus improving throughput). This column always contains a summary of the execution status of the parallel threads.
    - The number of records processed by the job. For example, if this is the BILLING job, the number of records shows the number of accounts for which bills were created.
    - The number of errors detected during the batch job (this information appears in red).
    - The average, minimum and maximum execution time per thread.

Copyright © 2024, Oracle and/or its affiliates.

- You can click on the hyperlink in the adjacent column to transfer to the Batch Run Tree where the details of the run's parallel threads and errors are shown.

Chapter 16

Workflow and Notifications

Topics:

- The Lifecycle Of A Workflow Process And Its Events
- Workflow Process Maintenance
- Workflow Event Exception
- Uploading Notifications
- Downloading Notifications

We use the term "notification" to reference the electronic transactions that you exchange with third parties when:

- They need information about a customer
- They need to change something about a customer

The workflow process contains workflow events. These events perform the processing necessary to execute the notification.

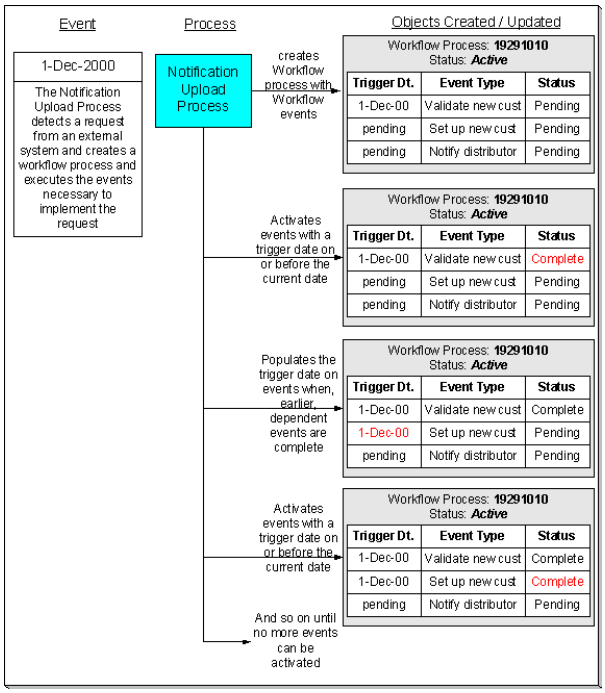
In addition to processing incoming notifications, workflow processes can be used to automate many different types of multi-event processes.

This section describes how to manage your workflow and notification processing.

Note: In the section, Defining Workflow and Notification Options, we describe how to set up the control tables that automate most of your notification and workflow processing. If you set these tables up properly, little or no operator intervention should be necessary. You should only have to access the workflow and notification pages when you need to override the standard procedures embodied in your workflow processes.

The Lifecycle Of A Workflow Process And Its Events

The following diagram illustrates the lifecycle of a typical workflow process and its workflow events:



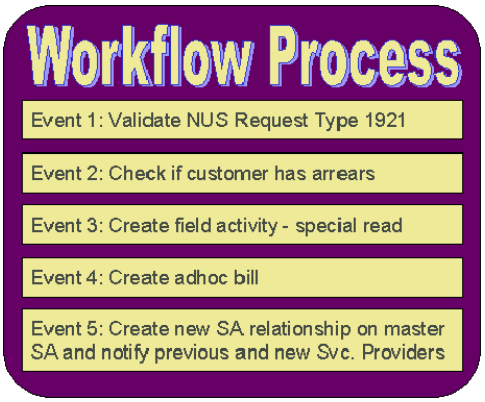
The above diagram is referenced in the following topics.

What Is A Workflow Process?

A workflow process is difficult to define because its flexible design can be used to automate many different types of multi-event processes. For example,

- You can use a workflow process to manage events that transpire in some designated market when a customer wants to switch financial service providers.
- You can use a workflow process to notify interested parties when something about a customer changes. For example, if a customer changes their phone number, a workflow process can send notifications to the customer's various service providers.

The following is an illustration of the workflow process:



Copyright © 2024, Oracle and/or its affiliates.

Note: Bottom line: A workflow process contains one or more workflow events. Each event is activated on its respective trigger date. When an event is activated, the system executes the activation algorithm associated with the event's event type. It's up to you to define exactly what algorithms should be associated with your workflow event types. Refer to Designing Workflow Event Types for sample events.

A Workflow Process Contains Relevant Data

Linked to a workflow process are the relevant data associated with the process. This information can be for audit purposes or it may be needed by one or more of the event algorithms linked to the process to do its work. For example,

- If a workflow process is associated with new service, many pieces of information are needed to be linked to the workflow including the address of the new service, the date service should start, perhaps the desired rate and perhaps person information, if this is for a new customer.
- If a workflow process is triggered when we receive a notification that we have a new customer, the notification is linked to the workflow process.

This data may be linked to the workflow process in one of two ways: as context or as characteristics. The following points describe each method:

- Context Collection
  - Uses a context type to identify the type of data. The values of the context type are stored in a lookup field and must be customized by your implementers.
  - The values are stored in the context value field. Values may be up to 50 bytes long.
  - The primary key of the record includes context type and context value so multiple values may be entered for the same type.
  - Note, the context value is a generic field. If the value of the field is a foreign key, there is no way for the system to enable drill down capability to this foreign key's entity.
- Characteristic Collection
  - Uses the characteristic type table to identify the type of data. Characteristic types may be pre-defined, ad-hoc, foreign key or file location.
  - The values are stored in different characteristic value fields depending on the type of characteristic. The ad-hoc and file location values may be up to 254 bytes long.
  - The primary key of the record includes characteristic type and a sequence number so multiple values may be entered for the same type.
  - If the characteristic is of type foreign key reference, a search is available for selecting an appropriate value and a drill down key is available for viewing the related record.

Note: The original design included only the context collection. The characteristic collection was added later as a result of limitations in the context functionality that was supported in other parts of the system by characteristic functionality. It is expected that characteristics will be the preferred method to use for implementations. Context will continue to be supported for earlier releases.

Information in a workflow process' context or characteristics collection is referenced by the workflow process' event algorithms when they are activated.

If a workflow process is created to process an incoming notification from a third party, the first event algorithm for your workflow process must populate the workflow process' context or characteristics with individual fields specified on the incoming notification.

Fastpath: Refer to Uploading Notifications Into The System for more information.

Copyright © 2024, Oracle and/or its affiliates.

Note: If an account ID in Oracle Revenue Management and Billing is linked to an `active` workflow process either via context or via a characteristic, you may configure your system to display an alert when that account is displayed in the current context. To display this alert, create one or more algorithms for the algorithm type CCAL-WF, setting the parameters as appropriate to display account and indicate whether or not the account will be found in the context or characteristic collection. Plug in your new algorithm(s) to the installation option - algorithm collection.

How Are Workflow Processes Created?

A workflow process may be created as follows:

- The Notification Upload background process (referred to by the batch code of `WFPRINIT`) creates a workflow process when a notification is interfaced into the system. The workflow process has one or more workflow event(s). Workflow processes created as a result of a notification upload staging record will reference the NUS ID that created it. Refer to The Big Picture Of Notification Processing for more information about notifications.
- An operator may create an ad hoc workflow process at will.
- If you customize the system accordingly, the system will create a workflow process when a significant event occurs. The purpose of this type of workflow process is to send outgoing notifications to other interested parties to apprise them of the event. Refer to Designing Notification Downloads for more information.

Whenever a workflow process is created, the system uses a workflow process *template* to create the workflow process. The workflow process template defines the various events that will transpire over the course of the process. Refer to Designing Workflow Process Templates for more information.

In the above diagram, the event on 1-Dec-00 shows the creation of a workflow process with three workflow events. Notice that the events all exist in the `Pending` state. The events will exist in this state until their trigger date is on or before the current date.

Fastpath: Refer to What Type Of Workflow Process Is Created? to understand how the Notification Upload Process creates workflow processes when notification requests are interfaced into the system. Refer to The Big Picture Of Workflow Events for more information about a workflow process's events.

How Are Workflow Events Completed?

Several background processes can cause a workflow event to complete as described below:

- A background process (referred to as `WFET`) activates events whose trigger date is no longer in the future. (An event's trigger date can be set to a future date when its earlier, dependent events are completed if the workflow process template so indicates). After it activates an event, this process also activates later dependent events until there are no more events to activate. Refer to Workflow Event Dependencies & Trigger Date for more information.

Note: WFET attempts to activate workflow events for a workflow process. The system orders the workflow process records by their workflow process ID, which is a system-generated random number. As a result, you should ensure that the activation algorithms do not expect WF process records to be processed in any other order, such as the order that the WF process records were created. For example, if a "drop request" and an "add request" are both received for the same customer, two workflow processes are created. The activation algorithms for the events for these two processes should work regardless of whether the "drop request" is processed first or the "add request" is processed first.

- The various waiting background processes are responsible for activating events when the thing they are waiting on is complete. Refer to Waiting Events And Their Waiting Process for more information.

In addition, if the workflow event type indicates that manual completion is allowed, a user may change the status of the event to `complete` using the workflow process user interface. When an event is manually completed, the system also activates later dependent events until there are no more events to activate.

Fastpath: Refer to The Big Picture Of Workflow Events for more information about the various types of workflow events and what happens when they are completed.

Copyright © 2024, Oracle and/or its affiliates.

How Are Workflow Processes (and their Events) Canceled?

The background process that is responsible for activating events (referred to as `WFET`) automatically cancels a workflow process when one of its events fails. Refer to Some Workflow Events May Fail for more information.

Refer to How Are Workflow Events Canceled for more information.

Besides these automatic cancellations, an operator may cancel a workflow process at will.

Workflow Process Maintenance

A workflow process is hard to define because its flexible design can be used to automate many different types of multi-event processes. For example,

- You can use a workflow process to manage the events that transpire in some designated market when a customer wants to switch financial service providers.

Fastpath: For more information about workflow processes, refer to The Lifecycle Of A Workflow Process And Its Events.

The topics in this section describe the pages on which workflow process-related information is maintained.

Workflow Process - Main

The Main page contains core workflow process information. Open this page using **Workflow and Notification, Workflow Process**.

Description of Page

The top portion of every page contains the following information:

Workflow Process is a concatenation of summary information about this workflow process. It is composed of the workflow process template's name and description, the workflow process's status, and the creation date of the process.

Workflow Process ID is the unique system-generated identifier for this workflow process.

The remaining fields are maintained on the main workflow process page.

The **Create Information** describes how the workflow process was created. If the workflow process was created by the system, the identity of the process, and related control table references are displayed. If the workflow process was created by a user, the message indicates this.

Workflow Status defines the state of the workflow process. The following values may exist:

`Active` The system creates a workflow process in the `Active` state. A workflow process remains in this state until the last event linked to the process is complete OR an event fails OR you want to cancel the process for whatever reason.

`Inactive` A workflow process becomes `Inactive` when the last event linked to the process is completed OR an event fails OR you want to cancel the process for whatever reason. The **Status Reason** field tells you how a workflow process became inactive.

Fastpath: To completely understand the significance of a workflow process's status, refer to The Lifecycle Of A Workflow Process And Its Events.

Status Reason defines how a workflow process became `Inactive` or why a workflow process is `Active`. The following values may exist:

`Canceled by User` This reason should be used when an operator manually inactivates a workflow process (for whatever reason).

Copyright © 2024, Oracle and/or its affiliates.

`Canceled by System` This reason is populated by the system when it cancels a workflow process.

`Events Pending` This reason is populated by the system when a process has pending events.

`Complete by System` This reason is populated by the system when it completes the last event linked to the workflow process.

`Complete by User` This reason is populated by an operator when they manually complete a workflow process.

Workflow Process Template defines the workflow events in the process. You can override these events on the **Events** page.

> **Note: Changing the template.** If you change the template when all events are `pending`, the system will remove the `pending` events and replace them with the new template's events.

> **Fastpath:** For more information, refer to Setting Up Workflow Process Templates.

If your event has been canceled by the system, a **Message** appears indicating the reason for cancellation.

Create Date/Time defines the start date/time of the workflow process. This is important as it affects the trigger dates of the process's workflow events. If you change this date after events exist, the system will NOT update the trigger date on the events. If you need to change the trigger date on the events, you can do this manually on the **Events** page OR you can change the Workflow Process Template to something (anything) and then change it back to the desired template. When you change the template, the system removes the events and creates new ones using the creation date / time on the process.

The workflow process was created by a **Notification Upload** staging record, information about the notification upload staging record is displayed.

Enter any **Comments** about the workflow process.

This bottom of this page contains a tree that shows the various objects linked to the workflow process. You can use this tree to both view high-level information about these objects and to transfer to the respective page in which an object is maintained.

Workflow Process - Char/Context

The Process Char/Context page lists the context collection and/or the characteristic collection associated with the workflow process. Open this page using **Workflow and Notification, Workflow Process, Process Char/Context**.

> **Fastpath:** Refer to A Workflow Process Contains Relevant Data for more information about context and characteristics.

Description of Page

Use characteristics and/or context to capture data to be used by your workflow process' event algorithms. Refer to *A Workflow Process Contains Relevant Data* for more information.

The **Characteristics** grid displays the **Characteristic Type** and **Characteristic Value** for each attribute that can be accessed by the workflow process' event algorithms.

The **Context** grid displays the Context Type and Context Value for each attribute that can be accessed by the workflow process's event algorithms.

> **Note:** The values for Context Type are customizable using the Lookup table. This field name is WF_PROC_CTXT_FLG.

Workflow Process - Events

The events page contains the activities that will be performed over the course of the workflow process. Open this page using **Workflow and Notification, Workflow Process, Events**.

Copyright © 2024, Oracle and/or its affiliates.

> **Note:** We strongly recommend that you understand the information described in How Are Workflow Events Completed? before using this page.

The number and type of events associated with a workflow process are defined by the process's workflow process template (which is defined on the first page). Refer to Setting Up Workflow Process Templates for more information.

Description of Page

The top portion of every page in this page contains the following information:

Workflow Process is a concatenation of summary information about this workflow process. It is composed of the workflow process template's name and description, the workflow process's status, and the creation date of the process.

Workflow Process ID is the unique system-generated identifier for this workflow process.

The scroll controls the display of the individual workflow events that comprise the workflow process. The grid shows the events that must be completed before the workflow event is triggered. Only one workflow event is displayed at a time.

Event Sequence is the unique identifier of the event.

Workflow Event Status defines the state of the event.

> **Fastpath:** For more information, refer to Workflow Event Lifecycle.

If the workflow event status is `Error`, then the message related to the error is displayed adjacent.

> **Note: Failed events are special.** If the workflow event's status is `Failed`, the system displays the **Failure Algorithm** that was executed (the type of failure algorithm is defined on the event's event type) and, if the failure algorithm created a workflow process, the system displays information about the **Created Workflow Process**.

The **Workflow Event Type** defines the event's activity.

> **Fastpath:** For more information, refer to Setting Up Workflow Event Types.

If **Dependent on Other Events** is turned on, then the event's trigger date is only determined after the event(s) on which it depends are completed. When this switch is on, a grid appears at the bottom of the page in which you define the event(s) on which this event depends.

> **Fastpath:** For more information about dependent events, see Workflow Event Dependencies & Trigger Date.

The **Trigger Date** defines the date when the system completes (i.e., executes) the event. This field is gray when the event is dependent on the completion of other events.

> **Fastpath:** For more information about a workflow event's trigger date, see Workflow Event Dependencies & Trigger Date.

Days After Prev ious **Response** defines how a dependent event's trigger date is calculated. The trigger date is set equal to `X` days after the last dependent event is completed, where `X` is the number in this field. This field is gray when the event is not dependent on other events.

> **Fastpath:** For more information about dependent events, see Workflow Event Dependencies & Trigger Date.

The **Status Change Date/Time** is date and time the event's status was last changed.

If the event can go into the `Waiting` state, the **Wait Date/Time** is populated with the date / time on which the event started waiting.

> **Fastpath:** Refer to Waiting Events And Their Waiting Process for more information.

The grid at the bottom of the page contains the events on which this event depends. This grid is only visible when the event is dependent on other events (as defined by the Dependent on Other Events switch).

Copyright © 2024, Oracle and/or its affiliates.

Workflow Process - Event Char/Context

The Event Context page lists the entity(ies) that were created as a result of the execution of a workflow event. The event algorithm may choose to record the entity using a characteristic or a context entry. Open this page using **Workflow and Notification, Workflow Process, Event Char/Context**.

Fastpath: Refer to *A Workflow Process Contains Relevant Data* for more information about context and characteristics.

Description of Page

Your event algorithms may record information that was created by the algorithm. The information may be recorded as a characteristic or in a context entry. This information is for audit purposes only.

If **Characteristics** are used to record entities created by the algorithm, this grid displays the **Characteristic Type** and **Characteristic Value** for each entity that was created as a result of the execution of workflow event.

If **Context** entries are used to record entities created by the algorithm, this grid displays the **Context Type** and **Context Value** for each entity that was created as a result of the execution of workflow event.

Note: The values for Context Type are customizable using the Lookup table. This field name is WF_EVT_CTXT_FLG.

Workflow Event Exception

As described in Workflow Events May Be In Error, a record is written the workflow event exception record for every workflow event record that is in error.

To view the messages associated with the exception records, schedule the `TD-WEXTL` background process. This process generates a To Do entry for every record in the workflow event exception table.

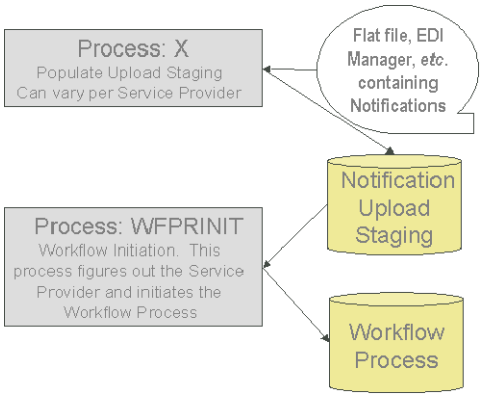
After correcting the cause of the error, drill into the Workflow Process - Events tab and change the status from `Error` to `Pending`. The next time the event activation process runs, it attempts to process this record again.

Uploading Notifications

This section describes how the system uploads notifications. Refer to Uploading Notifications Into The System for more information.

Notification Upload Background Process

The following diagram illustrates the processes involved in the uploading of notifications into the system.

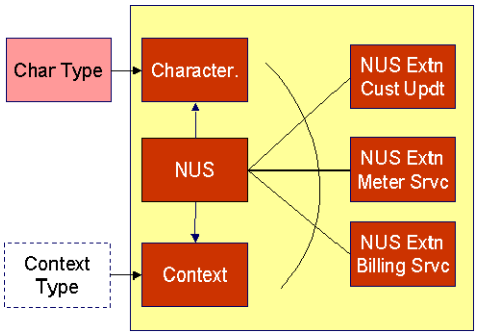


Copyright © 2024, Oracle and/or its affiliates.

The topics in this section describe how these processes work.

Process X - Populate Notification Upload Staging

Process X refers to the mechanism used by your organization to populate the various staging tables (shown in the yellow section of the following entity relationship diagram).



The NUS record contains only key header information, but it does not contain detailed information that would be required to process the notification. To define the other required information, like Account ID, you have three options:

- NUS Characteristics: This is a collection of characteristic values, where the characteristic type describes the data that is contained in the characteristic values. In order to set this up correctly, the appropriate characteristic types need to be defined.
- NUS Context: This is a collection of context values, where the context type describes the data that is contained in the context values. In order to set this up correctly, the appropriate context types need to be defined by your implementers.
- NUS Extension: This is a separate table, with explicit columns defining the data. In order to set this up correctly, a new table and new user interface will need to be created by your implementers, along with a new Extension type.

For any of the methods, the first Workflow Event in your workflow process will have the responsibility to retrieve the core information from the Characteristic collection, the Context collection or the Extension record and populate the workflow process context or characteristic collection accordingly.

The topics in this section describe each of these tables.

Notification Upload Staging

You must create an upload staging record for each notification being processed in the system. The name of this table is CI_NT_UP .

The following table describes each column in the CI_NT_UP table:

Column Name	Length	Req'd	Data Type	Comments
NT_UP_ID	12	Y	A/N	This is a sequential unique key. This key is referenced on the NUS extension record. (The NUS extension record contains almost all of the fields associated with the notification.) This key is a sequential number so you can use a database function to assign the value when populating the table.
NT_XID_CD	30	Y	A/N	This identifies the external system that sent the notification (e.g., a DUNS number). This must correspond with the prime key of one of your external systems. Refer

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Length	Req'd	Data Type	Comments
				to Designing External Systems for more information.
NT_UP_XTYPE_CD	30	Y	A/N	This identifies the transaction type of the notification. This must correspond with the prime key of one of your notification upload types. Refer to Designing Notification Upload Types for more information.
NT_UP_STAT_FLG	2	Y	A/N	This must be set to `P` for `Pending`.
NT_UP_EXTSN_FLG	4	Y	A/N	This identifies the type of notification upload staging extension record that contains the details of this notification upload. Valid values: `NONE`, `SPL`. Use `NONE` if you will use the context collection for your related data.
CRE_DTTM	26	Y	DateTime	This is the date / time on which the notification upload staging record was created. This must be populated with the current date / time.

Notification Upload Staging Context

If desired, you may insert rows into the upload staging context collection. The context collection enables you to define all the core information associated with the upload staging record. Context flag values will need to be defined for every piece of data you wish to capture. This option may be used in favor of the extension record option, defined below, because you do not need to define new database tables or new user interfaces for each new notification upload type.

The name of this table is CI_NT_UP_CTXT.

The following table describes each column in the CI_NT_UP_CTXT table:

Column Name	Length	Req'd	Data Type	Comments
NT_UP_ID	12	Y	A/N	This is the unique key of the NUS record to which this collection is linked.
NT_UP_CTXT_FLG	4	Y	A	This field describes the data that is stored in the Context Value. Refer to Notification Upload Staging for more information.
CTXT_VAL	50	Y	A	This is the value of the context.

Notification Upload Staging Characteristics

If desired, you may insert rows into the upload staging characteristics collection. The characteristics collection enables you to define all the core information associated with the upload staging record. Characteristic types will need to be defined for every piece of data you wish to capture. This option may be used in favor of the extension record option, defined below, because you do not need to define new database tables or new user interfaces for each new notification upload type.

The name of this table is CI_NT_UP_CHAR .

The following table describes each column in the CI_NT_UP_CHAR table:

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Length	Req'd	Data Type	Comments
NT_UP_ID	12	Y	A/N	This is the unique key of the NUS record to which this collection is linked.
CHAR_TYPE_CD	8	Y	A	This field describes the data that is stored in the characteristic value field. Refer to characteristic type for more information.
SEQ_NUM	3	Y	N	This value is used to differentiate multiple characteristics of the same type. Values are typically incremented by 10.
CHAR_VAL	16	N	A	Populate this field if your characteristic type is `predefined`.
ADHOC_CHAR_VAL	254	N	A	Populate this field if your characteristic type is `ad-hoc` or `file location`.
CHAR_VAL_FK1-CHAR_VAL_FK5	50 each	N	A	Populate these fields if your characteristic type is `foreign key reference`. Up to five columns of 50 bytes each are provided to accommodate compound keys.

Notification Upload Staging Extension - Base Sample

A notification upload staging EXTENSION record may be used in lieu of Context to contain the core information associated with a notification upload record. This option enables you to be explicit in your definition of the fields that must be captured for each notification upload type.

The sample extension record that is supplied with the system is just that - a sample. If you use this option, you will need one such extension record for each different type of notification upload record layout you support. It's important to understand that all of your different notification upload extension record types must reference the *core* Notification Upload Staging record because the system uses the *core* record to drive the notification upload process.

The name of the *sample* notification upload extension record is CI_NT_UP_XSPL .

The following table describes each column in the CI_NT_UP_XSPL table:

Column Name	Length	Req'd	Data Type	Comments
NT_UP_ID	12	Y	A/N	This is the reference to the Notification Upload Staging record that this record is an extension of.
REC_DTTM	26	Y	Date/Time	This is the date / time the record was received.
RESPONSE _CODE	1	Y	A/N	This column is used when the notification is an indication of the acceptance or a rejection of an earlier notification request. If specified, this must be either `A`- accept or `R`- reject.
SDR_ACCT_ID	20	Y	A/N	This is the customer's account ID in the sender's system.
RCV_ACCT_ID	20	Y	A/N	This is the customer's account ID in the receiver's system.
SDR_REC_ID	16	Y	A/N	This is a transaction identifier supplied so responses to this record may be identified.

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Length	Req'd	Data Type	Comments
				The contents of this field will be returned on notification download records that respond to this notification upload record.
RCV_REC_ID	16	Y	A/N	This is the transaction identifier received in a previous request record. This field is only populated if the notification being uploaded is an acceptance / rejection of a previously sent notification download record.
BILL_OPT_FLG	4	Y	A/N	This defines the billing relationship between the service provider and your organization. This value is used in conjunction with BILL_CALC_OPT_FLG to determine the service provider associated with the NT_XID_CD on the CI_NT_UP record. Valid values are: `DUAL`, `ESP`, `UDC`.
BILL_CALC_OPT_FLG	4	Y	A/N	This defines the bill calculation method used when your organization provides billing services for the service provider. This value is used in conjunction with BILL_CALC_FLG to determine the service provider associated with the NT_XID_CD on the CI_NT_UP record. Valid values are: `BILL` (bill ready), `RATE` (rate ready).
POSTAL_CD	12	Y	A/N	The service address's postal code. This is used as a crosscheck to ensure the notification is referencing the correct customer.

Process WFPRINIT - Initiate Workflow Processes To Execute Notification Upload Staging Records

This process creates a workflow process for all Notification Upload Staging records in the `Pending` state. It determines the workflow process using a combination of the External Id and the NUS Type. Refer to What Type Of Workflow Process Is Created for more information. It stamps the NUS ID onto the newly created workflow process.

If the External ID on the staging record is not a valid External ID in our system, rather than creating an error, this process will find a workflow process template with a notification condition of `Unknown Notification ID` and create a workflow process. If the NUS Type on the staging record is not a valid NUS Type in our system, rather than creating an error, this process will find a workflow process template with a notification condition of `Unknown Notification ID` and create a workflow process. Refer to Setting Up Workflow Process Templates for more information. For these cases, the status of the notification upload staging record will be set to `Complete`.

Any errors detected during this process will be written to the Notification Upload Exception table. You can fix errors using the Notification Upload Staging page (don't forget to change the record's status back to `Pending`).

Fastpath: Refer to Uploading Notifications Into The System for more information.

Copyright © 2024, Oracle and/or its affiliates.

Validation of Data Related to the Notification Upload Staging Records

As mentioned above, the sole purpose of the Initiate Workflow Process (WFPRINIT) is to successfully create a workflow process. Once a workflow process is created, the NUS status is set to `Complete`.

At this point, neither the extension record, the context collection nor the characteristics collection have been processed. It is the responsibility of the first event in the workflow process to validate the fields in the extension record, the context collection and/or the characteristics collection. This algorithm will decide what to do when a problem is detected with the data. Some possible options are

- Set the algorithm status to `Fail`. Use this option when a critical error is detected, which probably cannot be fixed by a user. This will cause the remaining events to be canceled.
- Set the algorithm status to `Error`. Use this option when an error is detected, which can be corrected by a user. This will cause an exception record to be created. The user should be able to correct the source data in the extension record, the context collection or the characteristics collection, change the status back to `Pending` and rerun the Workflow Event Activation process. Note that if the validation algorithm uses this option, the ability to rerun this event should be handled by the coding of this algorithm. It should prevent duplicate logic from being performed.
- Send a To Do list entry and continue the workflow process by setting the algorithm status to `Complete`. Use this option when someone should be alerted of an unusual condition, but the process should not be halted. For this option, the To Do Type and User to use for posting the To Do entry should be parameters to the algorithm.

Note: We said that a possible correction for the validation workflow event in the `Error` is to modify data in the extension record, the context collection or the characteristics collection. You may find this a bit unusual because the parent NUS record is in the status of `Complete`. Remember that this NUS status indicates whether or not the system was able to successfully create a workflow process. There is no status for the data related to the NUS - only the status of the event, which validates it.

Notification Upload Staging Maintenance

The Notification Upload Staging page is used to view notification upload records.

Fastpath: Refer to Process X - Populate Notification Upload Staging for more information about these records.

Notification Upload Staging - Main

Use this page to define common attributes of notification upload staging records. Open this page using **Workflow and Notification, Notification Upload Staging, Main**.

Description of Page

The **Upload Staging ID** is the unique identifier of the notification upload staging (NUS) record.

Status shows the state of the NUS record. Potential values are: `Pending`, `Complete`, `Error`. To have the system reprocess a record that's in `Error`, change the status back to `Pending`. The system will revalidate the record the next time the notification upload process executes.

Create Date and **Time** are just that.

External System identifies the sender of the notification.

Notification Upload Type identifies the transaction type of the notification.

Extension identifies the type of notification upload staging extension record that contains the details of this notification upload. Valid values: `No Extension`, `Sample Extension Record`. You would use `No Extension` when you plan to define the data related to the NUS record in the characteristic collection or in the context collection rather than in an extension table. If your notification upload type indicates a navigation option associated with your extension record, a drill down button is enabled adjacent to this field.

Note: **Default Note**. The extension value defaults from the value specified on the notification upload type.

Copyright © 2024, Oracle and/or its affiliates.

> **Note:** The values for this field are customizable using the Lookup table. This field name is NT_UP_EXTSN_FLG.

If a **Workflow Process** has been created for this NUS record, information about the workflow process will be displayed.

This bottom of this page contains a tree that shows the various objects linked to the notification upload staging record along with information about the workflow process, if one has been created for the NUS. You can use this tree to both view high-level information about these objects and to transfer to the respective page in which an object is maintained.

Notification Upload Staging - Characteristics/Context

Use this page to either characteristic records or context records to define data related to the NUS record. Either collection may be used as an alternative to defining an extension record. Refer to Process X - Populate Notification Upload Staging for more information.

Open this page using **Workflow and Notification, Notification Upload Staging, Characteristics/Context**.

Description of Page

The **Upload Staging ID** is the unique identifier of the notification upload staging (NUS) record.

The **Characteristics** collection enables you to define a collection of **Characteristic Types** and **Characteristic Values** defining data related to the NUS record.

The **Context** collection enables you to define a collection of **Context Types** and **Context Values** defining data related to the NUS record.

> **Note:** The values for this field are customizable using the Lookup table. This field name is NT_UP_CTXT_FLG.

> **Note:** You will notice that both the characteristic collection data and the context collection data may be modified even when the parent NUS record is `Complete`. This is to enable a user to correct problems with this data as detected by the validation event. Refer to Validation of Data Related to the Notification Upload Staging Records for more information.

Notification Upload Staging - Sample Upload Extension Record

The Notification Upload Staging page is used to maintain notification upload records.

> **Fastpath:** Refer to Process X - Populate Notification Upload Staging for more information about these records.

Open this page using **Workflow and Notification, Sample Upload Extension**.

Description of Page

Receive Date and **Time** are just that.

Response Code is used when the notification is an indication of the acceptance or a rejection of an earlier notification request. If specified, this must be either `Accept` or `Reject`.

Sender's Customer's Account ID is the customer's account ID in the sender's system.

Receiver's Customer's Acct ID is the customer's account ID in your system.

Sender Record ID is a transaction identifier supplied so responses to this record may be identified. The contents of this field will be returned on notification download records that respond to this notification upload record.

Receiver Record ID is the transaction identifier received in a previous request record. This field is only populated if the notification being uploaded is an acceptance / rejection of a previously sent notification download record.

Billing Option defines the billing relationship between the service provider and your organization. This value is used in conjunction with **Billing Calculation Option** to determine the service provider associated with the sender's external system (stored on the related notification upload staging record). Valid values are: `Dual Billing`, `ESP Consolidated`, `UDC Consolidated`.

Copyright © 2024, Oracle and/or its affiliates.

Billing Calculation Option defines the bill calculation method used when your organization provides billing services for the service provider. This value is used in conjunction with **Billing Option** to determine the service provider associated with the sender's external system (stored on the related notification upload staging record). Valid values are: `Bill Ready`, `Rate Ready`.

Postal Code is the service address's postal code. This is used as a cross check to ensure the notification is referencing the correct customer.

> **Note:** You will notice that the extension data may be modified even when the parent NUS record is Complete. This is to enable a user to correct problems with this data as detected by the validation event. Refer to Validation of Data Related to the Notification Upload Staging Records for more information.

Notification Upload Exception

A record is written the notification upload exception record for every notification upload staging record that is in error.

To view the messages associated with the exception records, schedule the TD-NTUPL background process. This process generates a To Do entry for every record in the notification upload exception table.

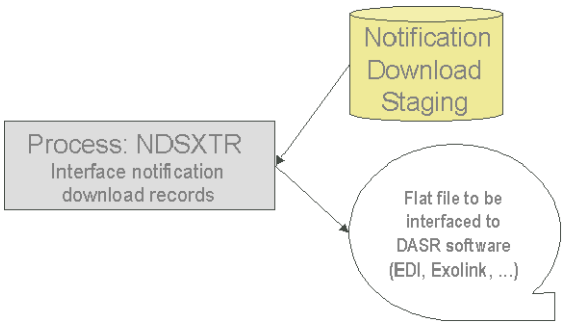
After correcting the cause of the error, drill into the Notification Upload Staging page and change the status from `Error` to `Pending`. The next time the notification upload process runs, it attempts to process this record again.

Downloading Notifications

This section describes how the system downloads notifications. Refer to How Are Notifications Sent Out Of The System? for more information. Information may be extracted from the system using a batch process, which runs in the background or using the XAI tool, which continuously searches for new NDS records and builds XML documents to send to an external recipient.

Notification Download Background Processes

The following diagram illustrates how a batch process, which runs in the background may be used to route notifications out of the system.



The topics in this section describe how this process works.

Process NDSXTR - Interface Notification Download Records (out of the system)

This process reads all notification download staging records marked with a given batch control ID & run number and creates the flat file that's passed to the outgoing notification software. Every NDS record references the recipient service provider and a notification download type. This process uses the service provider's notification download profile to determine the format of records interfaced out of the system. Refer to Designing Notification Downloads for more information.

Copyright © 2024, Oracle and/or its affiliates.

> **Note: This process is rerunnable.** You can reproduce the flat file at any time. Simply request this job and specify the run number associated with the historic run.

If you require different routing methods, you will need to create additional versions of this program.

Notification and XAI

NDS records may also be processed using the XAI tool.

> **Fastpath:** Refer to Notification and XAI for more information.

NDS Contains Interesting Data

The NDS record also contains interesting data

- The main use for this is to indicate the workflow process and event that created the NDS
- The process that extracts the data may then find the other data related to this NDS by looking at the workflow process context or characteristics

This interesting data may be stored in one of two ways

- Through NDS context collection. Use this method if your NDS is interfaced to the outside world using XAI. Refer to Designing Notification Download Types for more information.
- Through NDS characteristics collection. For NDS records that are interfaced through a batch process, it is expected that using characteristics is the preferred method because if your characteristic references a foreign key, your user can easily drill down to the related record.

> **Note:** The algorithms provided with the base product use the context collection to populate the interesting data for an NDS.

Notification Download Staging

The Notification Download Staging page is used to view notification download records. This is especially useful if you send out a notification and request a response.

> **Fastpath:** Refer to Notification Download Background Processes for more information about these records.

Notification Download Staging - Main

Open this page using **Workflow and Notification, Notification Download Staging**.

Description of Page

The **Download Staging ID** is the unique identifier of the notification download staging (NDS) record.

Service Provider is the service provider to whom this download staging record will be sent. The description of the service provider is displayed adjacent.

Notification Download Type indicates the reason for this notification. The description is displayed adjacent.

> **Fastpath:** Refer to Designing Notification Download Types for more informationabout notification download types.

The **Processing Method** indicates whether this record will be processed by a `Batch` extract process or through the `XAI` tool. The processing method defined on the Notification Download Profile will be used to populate this value.

Copyright © 2024, Oracle and/or its affiliates.

If the processing method is `Batch`, then **NDS Extract Process** indicates the process that will extract this record. This value is populated based on the service provider's notification download profile. **NDS Extract Run** indicates in which batch run this record was extracted or will be extracted.

> **Fastpath:** Refer to Designing Notification Download Profiles for more informationabout the NDS Extract Process.

The **Retry Count** is used by the XAI tool to keep track of how many times the tool tried to process this record and could not process the record, resulting in an error.

The **Create Date/Time** indicates the date and time that this record was created.

Status defines the state of the NDS record. This is only used when the processing method is `XAI` and will be protected when the processing method is `Batch`. Refer to Lifecycle of Notification Download Staging for more information.

This bottom of this page contains a tree that shows the various objects linked to the notification download staging record. You can use this tree to both view high-level information about these objects and to transfer to the respective page in which an object is maintained.

Notification Download Staging - Characteristics/Context

This page displays the characteristics and/or context associated with the download staging record. Refer to NDS contains interesting data for more information.

Open this page using **Workflow and Notification, Notification Download Staging, Characteristics/Context**.

Description of Page

The **Download Staging ID** is the unique identifier of the notification download staging (NDS) record.

If **Characteristics** are used by the algorithm that created this record, to record information related to the record, this grid displays the **Characteristic Type** and **Characteristic Value** for each entity associated with the download staging record.

If **Context** entries are used by the algorithm that created this record, to record information related to the record, this grid displays the **Context Type** and **Context Value** for each entity associated with the download staging record.

> **Note:** The values for **Context Type** are customizable using the Lookup table. This field name is NT_DWN_CTXT_FLG.

Notification Download Staging - XAI Download Staging

Open this page using **Workflow and Notification, Notification Download Staging, XAI Download Staging**.

Description of Page

For XAI messages, an NDS Type may point to multiple XAI Route Types. Each route type is linked to an XAI sender. The system keeps track of each routing and each response. This collection of **XAI Route Type** includes the **XAI Download Status** and **Message**. To look at more detail, drill down into a specific XAI Routing and you are transferred to the XAI Download Staging page, where you may view the XML Request and XML Response.

Notification Download Exception

A record is written the notification download exception record for every notification download staging record that is in error. This is only applicable to records with a processing method of `XAI`.

To view the messages associated with the exception records, schedule the `TD-NTDWN` background process. This process generates a To Do entry for every record in the notification download exception table.

> **Fastpath:** Refer to Lifecycle of Notification Download Staging for more information.

Copyright © 2024, Oracle and/or its affiliates.

Chapter 17

Overdue Financial Obligations

Topics:

- Overdue Processing Background Information
- Overdue Process Maintenance
- Payment Arrangement for Bills

The system periodically monitors how much your customers owe to ensure they haven't violated your overdue rules. When a violation is detected, the system creates an overdue process. The overdue process contains the events meant to prod the customer to pay (e.g., letters, ToDo entries, write-off outstanding debt, etc.). This section describes how to manage your overdue (i.e., collections) processing.

Note: Collecting on unpaid bills. The overdue processing module has been designed to collect on virtually anything from an unpaid bill to an unmatched financial transaction. You tell the system what you collect on by configuring the various overdue processing control tables. In this release, the base-package is delivered with the algorithms that support collecting on overdue bills. If your organization practices balance-forward accounting (i.e., collection is based on overdue contract balances), you will not use this functionality. Rather, you will use the functionality described under Credit and Collections.

Overdue Processing Background Information

In the section, The Big Picture Of Overdue Processing, we describe how overdue processing works and how to set up the control tables that automate most functions. You will find that many of your questions regarding when and how overdue processes are created and canceled are described in this section.

Overdue Process Maintenance

An overdue process is a series of events (e.g., letters, ToDo entries) meant to encourage an account to pay its delinquent debt. The topics in this section describe the pages on which overdue processes are maintained.

Fastpath: Refer to The Big Picture Of Overdue Processes for background information.

Overdue Process - Main

The Main page contains core overdue process information. Open this page using **Delinquency Management, Overdue Process** from the main menu.

Description of Page

Note: Refer to How To Perform Common Overdue Process Functions for more instructions describing how to use this page.

Overdue Process contains a concatenation of important information about the process. **Overdue Process ID** is the system-assigned unique identifier of the process. These values only appear after the overdue process is added to the database.

Note: Summary information may be overridden. Refer to Overdue Process Information Is Overridable for how your implementation can override the summary information that appears throughout the system.

Account ID identifies the overdue process's account.

Note: Multiple overdue processes may be linked to an account. It's important to be aware that it's possible for multiple, active overdue processes to be linked to an account. The Alert zone in the dashboard contains a summary of the account's overdue processes.

Status defines the state of the overdue process. The following values may exist:

- `Active, Events Pending.` Overdue processes are initially created in this state. An overdue process remains in this state until there are no `Pending` or `Waiting` events.
- `Inactive, Canceled by User.` An overdue process will exist in this state when it's been manually canceled by a user. Navigate to the **Log** tab to see when this happened and who did it.
- `Inactive, Canceled by System.` An overdue process will exist in this state when it's been canceled by the system (typically because the overdue obligations were satisfied). Navigate to the **Log** tab to see when this happened.
- `Inactive, Completed.` An overdue process will exist in this state when the system completes its last event. Navigate to the **Log** tab to see when this happened.

The **Cancel** button appears if the overdue process is `Active, Events Pending` and the process's **Overdue Process Template** has a `Cancel Logic` plug-in. Click this button to cancel the process.

The **Trigger Events** button appears if the overdue process is `Active, Events Pending`. Click this button to activate all pending events that are ready for activation.

Copyright © 2024, Oracle and/or its affiliates.

> **Note: If you can't wait for the Overdue Event Manager to run.** The Overdue Event Manager is a background process that activates events on their trigger date. If you don't want to wait for this process to run, you can click the **Trigger Events** button on the **Main** tab to activate the process's events (that are ready for activation).

Overdue Process Template defines the template that was used to create the overdue process's events. You can override these events on the **Events** page. This field is unprotected when the process is `Active` and all events are in either the `Pending` or `Canceled` states.

> **Note: Changing the template.** If you change the template, the system will delete the events and replace them with the new template's events.

Start Date/Time defines the start date/time of the overdue process. This field is protected after the overdue process is saved on the database. This field is used to derive the trigger date of some overdue events when the process is first created. Subsequent changes to the start date change the trigger dates on the events accordingly.

Inactive Date/Time is the date and time that the overdue process became `Inactive`. This field is hidden if the process is `Active`. You can see more details about when and how the inactivation occurred by navigating to the **Log** tab.

Base Date is reserved for future releases. This field is protected when the overdue process is `Inactive`.

Enter any **Comments** about the overdue process. This field is protected when the overdue process is `Inactive`.

The **Collecting On** grid holds the overdue objects. For example, if this process manages overdue bills, the grid contains a list of the bills being collected on by this process. This grid is unprotected when the process is `Active`. The following points describe the columns that appear in the grid:

- Press the + button to add a new overdue object to the process.
- Press the - button to remove an overdue object to the process.
- The next column contains the unique identifier of the overdue object. For example, if this process manages overdue bills, this column will contain the bill ID. The type of object that's specified in this column is control by the overdue process template.
- **Original Amount** contains the original amount of the overdue object.
- **Unpaid Amount** contains the unpaid amount of the overdue object.

The **Tree** at the bottom of the page shows a variety of information about the overdue process including its events. You can click the tree nodes to navigate to the page on which the related object is maintained.

Overdue Process - Events

This page contains the activities that are performed to persuade the customer to pay the outstanding debt. Open this page using **Delinquency Management, Overdue Process** and then navigate to the **Events** tab.

Description of Page

> **Note:** Refer to How To Perform Common Overdue Process Functions for more instructions describing how to use this page.

Refer to the first tab for a description of **Overdue Process** and **Overdue Process ID**.

The **Overdue Events** scroll contains the process's overdue events. Refer to Overdue Processes Are Created From Templates for information about how the system defaults a process's events from its template.

- Note, all information in the scroll is protected if the event's Status is `Complete` or `Canceled`.
- **Event Sequence** is the unique identifier of the event.
- **Status** defines the state of the event.
- The **Cancel** button appears if the event is `Pending` or `Waiting`, and the event's **Overdue Event Type** has a `Cancel Logic` plug-in. Click this button to cancel the event.
- The **Overdue Event Type** defines the event's activity (e.g., Email sent, a ToDo entry is generated, a letter is sent).
- The remaining fields work in unison:

Copyright © 2024, Oracle and/or its affiliates.

- - If **Dependent on Other Events** is turned on, the event can only be triggered after the events specified in the **Event Dependencies** grid are all `Complete` or `Canceled`. You use the **Days After** field to define when this event is activated. For example, if you enter 5, this event will be activated 5 days after all of the dependent events are completed / cancelled. If you enter 0, this event will be activated immediately after the dependent events are completed / cancelled. Refer to Calendar vs Work Days for a descriptions of how days are counted.

If all dependent events are already `Completed` or `Canceled`, you cannot specify **Days After**. Rather, you must enter the desired activation date in the **Trigger Date** field.

- If **Dependent on Other Events** is turned off, enter the desired activation date in the **Trigger Date** field. If this field is off, the **Days After** field and the **Event Dependencies** grid are hidden.

Note: If you can't wait for the Overdue Event Manager to run. The Overdue Event Manager is a background process that activates events on their trigger date. If you don't want to wait for this process to run, you can click the **Trigger Events** button to activate the process's events (that are ready for activation).

Overdue Process - Log

This page contains log entries that highlight significant events in the process's life. Open this page using **Delinquency Management, Overdue Process** and then navigate to the **Events** tab.

Description of Page

Please note the following about the entries in this grid:

- The system automatically creates log entries when significant events occur. Refer to Overdue Log for more information.
- You can manually add a log entry to an Active process by pressing the + button and enter the **Details**. You cannot modify or delete this information about saving it.
- The following information appears in the grid:
  - **Date/Time** contains the date and time the log entry was created.
  - **Details** contain the user-specified or system-generated information about the log entry.
  - **Relate Object** is populated on log entries that were created to record the creation of some other object. For example, if an overdue event creates a customer contact, the related object contains information about the customer contact. Please note that if the object description is shown in blue, you can click on the object's description to drill down to the object.
  - **Related Process / Event** appears if the log entry was created by an overdue event. It references the unique identifier of the process / event.
  - **Log User** contains the user who caused the log entry to be created.
  - **Log Type** indicates how the log entry was created. The possible values are:
    - `User.` A user manually created this entry.
    - `System.` This system created this entry.

Overdue Process - Characteristics

The **Characteristics** tab allows you to define characteristics for an overdue process. It contains the following fields in a grid:

Field Name	Field Description
Effective Date	Used to specify the date from when the characteristic is effective for the overdue process.
Characteristic Type	Used to indicate the characteristic type.

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description
	Note: The list includes only those characteristic types where the characteristic entity is set to **Overdue Process**.
Characteristic Value	Used to specify the value for the characteristic type.
	Note: When you specify the value for a predefined characteristic type, the description of the characteristic value appears corresponding to the **Characteristic Value** field. When you select a predefined characteristic type, the **Search** icon appears corresponding to the **Characteristic Value** field. On clicking the **Search** icon, you can search for a predefined characteristic value.

If you want to define more than one characteristic for the overdue process, click the **Add** (+) icon and then specify the details. However, if you want to remove a characteristic from the overdue process, click the **Delete** (🗑) icon corresponding to the characteristic.

How To Perform Common Overdue Process Functions

The topics in this section describe how to perform common overdue process maintenance functions. Refer to The Big Picture Of Overdue Processing for high-level information about overdue processing.

How To Create An Overdue Process

99.9% of all overdue processes are created by the Overdue Monitor and require no human intervention. The other 0.1% are created by users on-line / real time. The following points describe how to create the 0.1%.

- Use **Control Central** to choose the account that requires a new overdue process.
- After the account is populated on Control Central, choose the **Overdue Process +** option on the account context menu to transfer to the overdue process transaction in add mode for the account.
- After the Overdue Process - Main page appears, specify the appropriate **Overdue Process Template**. The template is used to default the process's events (the template also controls numerous business rules, for example, when it will be canceled, how it will be canceled, etc). Refer to How To Change Overdue Events for a description of how you can override these events.
- Enter the overdue object(s) in the **Collecting On** grid. You must define at least one object.
- Save the overdue process.

How To Change Overdue Events

When an overdue process is first created, it has one or more overdue events. The events are the activities that will be performed to persuade the customer to pay the outstanding debt.

The number and type of events that are created when an overdue process is initiated are defined on the overdue process's overdue process template. The following points describe how to add / change / delete events on an overdue process if the defaulted events are not satisfactory.

- Use **Control Central** to choose the account with the overdue process whose events need to be changed.
- After the account is populated on Control Central, choose the **Overdue Process** option on the account context menu to transfer to the overdue process transaction in update mode for the account. Note, an account's `Active` overdue processes can be selected from the Alerts zone.
- To add a new event, transfer to the Overdue Process - Events tab and press the + button in the **Overdue Events** scroll. At this point, the event has not been added to the database; rather, it just exists in memory. Before you add the event to the database, you must specify the following information:

Copyright © 2024, Oracle and/or its affiliates.

- Choose an **Event Sequence** so that the new event will be positioned properly in respect of the other events.
- Choose a **Status** of `Pending`.
- Choose the desired **Overdue Event Type**.
- If the activation of the new event is dependent on the successful completion of earlier events, turn on **Dep on Other Event** and then:
  - specify the sequences of the dependent events in **Event Dependencies** and
  - specify the how many days after the completion of the last dependent event that the new event should be triggered.
- If the activation of the new event is NOT dependent on the successful completion of earlier events, turn off **Dep on Other Event** and use **Trigger Date** to define the date on which the event should be activated (i.e., completed).
- To delete an existing event, click on the event in the tree on Overdue Process - Main. This will transfer you to the **Events** tab where you can click the - button to remove the event. At this point, the event has not been removed from the database; rather, it's been removed in memory.
- To change an existing event, click on the event in the tree on Overdue Process - Main. This will transfer you to the **Events** tab where you can make the desired changes.
- After all desired changes have been made, save the overdue process.

How To Cancel An Overdue Process

The system will cancel an overdue process when the cancel criteria defined on the overdue process are satisfied. The following points describe how to manually cancel an overdue process.

- Use **Control Central** to choose the account with the overdue process to be cancelled.
- After the account is populated on Control Central, choose the **Overdue Process** option on the account context menu to transfer to the overdue process transaction in update mode for the account. A list of all overdue processes associated with the account appears. If only one overdue process exists, it is automatically selected for you. Note, an account's `Active` overdue processes can be selected from the Alerts zone.
- On the **Main** tab, click Cancel and answer any prompts.

Payment Arrangement for Bills

A payment arrangement is an agreement with a customer to payoff severely overdue debt in installments. Bills sent to customers with payment arrangements contain charges for both their current services and their payment arrangement installment amount.

Organizations that practice open-item accounting who collect on overdue bills may allow customers to pay off their overdue bills in installments by setting up a payment arrangement for one or more bills. This transaction is used to maintain bill-oriented payment arrangements.

Fastpath: Refer to Bill-Oriented Payment Arrangements for background information.

Open this page using **Delinquency Management, Payment Arrangement for Bill** from the main menu.

Description of Page

The **contract** (contract) **Info** is a concatenation of important details about the payment arrangement contract (PA contract). The **contract ID** is the unique identifier of the PA contract. These values only appear after the PA contract exists on the database.

The **Current Balance** that appears beneath **Contract Info** contains the PA contract's current balance. When you initially set up a PA contract this balance will be zero. It is only non-zero if the customer has not paid a billed installment.

Payoff Balance beneath **Contract Info** contains the PA contract's payoff balance. The payoff balance is the total amount to be paid off over the PA contract's life. It is only non-zero after delinquent funds have been transferred to the PA contract. This balance is only displayed when it differs from the **Current Balance**.

Copyright © 2024, Oracle and/or its affiliates.

Fastpath: Refer to Current Amount versus Payoff Amount for more information.

Account ID is the account associated with the PA contract.

The occurrences of **Current** and **Payoff Balance** beneath **Account** are the *account's* respective balances. These values are displayed to help you confirm exactly how much the customer currently owes versus how much they will have to payoff over time (remember, after you transfer funds to a payment arrangement, the amount transferred is reduced from the account's current balance).

The **Unpaid Bills** grid contains a row for each of the account's unpaid bills.

Adjacent to each Unpaid Bill is its **Unpaid Amount** and a check box used to select the bill. Every selected bill's unpaid amount will be transferred to the PA contract when you click the **Create** button.

CAUTION: The system automatically indicates that all unpaid debt should be transferred to the payment arrangement. It does this by checking each bill by default.

Total Candidates is the sum of ALL unpaid bills (not just the selected bills).

New Payoff Balance contains the PA contract's payoff balance. This is the total amount of debt that will be paid off over the lifetime of the PA contract. This value equals the current payoff balance on the PA contract plus all selected debt from the **Candidate Contracts** scroll.

The next fields work in unison.

- If you want to define the number of future bills that the customer has to payoff the selected unpaid amount, enter the number of **Installments**. When you tab out of the field, the system sets the **Arrange Amount** equal to **New Payoff Balance**/ **Installments**.

Note: Arrange Amount may not add up. You will note that the **Arrange Amount** is rounded up so that if you multiply the number of installments by the calculated installment amount, the result will be greater than the PA contract's payoff amount. Don't worry - only the actual dollar amount of the PA contract's total debt will be billed by the system. The last installment will be slightly lower than the other installments.

- If you want to define the amount charged on future bills to payoff the unpaid amount, enter the **Arrange Amount**. For example, if the customer agrees to pay off their delinquent debt by paying an additional $50 on each bill, you'd enter "50" in this field. Note - If you enter a dollar amount in this field and tab out, the system will calculate the number of installments and display the number in the **Installments** field.

Define the **Division** and **Contract Type** of the payment arrangement contract.

Note: Payment arrangement contract types. There is a field on contract type called Special Role Flag. Only those contract types with a role of `Payment Arrangement` may be selected.

Clicking the **Create** (or **Change**)button causes the system to perform the following:

- It creates / updates a payment arrangement contract (PA contract). It sets the installment amount on this contract equal to the installment amount specified on the page.
- It transfers unpaid debt from the selected bills to the PA contract. The adjustment type used to transfer these funds is defined on the PA contract's contract type.
- It links the "transfer from" adjustments to the unpaid financial transactions on the bills. This will cause the unpaid FT's to be linked to `Balanced` match events. This means that any `Active` Overdue Processes for the bills should `Inactivate` (i.e., cancel) because the bills will no longer be consider unpaid. Refer to How Are Overdue Processes Cancelled for more information..
- It reduces the current balance on the PA contract by the amount of debt transferred to the PA contract. Why? Because when delinquent debt is transferred to the PA contract, its current balance increases. Because current balance contains the amount the customer currently owes, this balance must be reduced because the customer is going to pay off the debt in installments (and therefore they don't currently owe anything). The adjustment type used to transfer these funds is defined on the PA contract's contract type.

Copyright © 2024, Oracle and/or its affiliates.

Clicking **Break** causes the PA contract to become broken. Please refer to the note below for what transpires when a PA is broken.

> **Note: Break logic is in a plug-in.** Please be aware that the logic that is executed when a payment arrangement is broken exists in a plug-in algorithm (plugged-in on the account's Collection Class Overdue Rules). The base package algorithm performs many functions. The bottom line is that when a PA contract is broken, the originating bills will have their debt reinstated and this will again be subject to the Overdue Monitor. Because this logic is in a plug-in, you can develop alternative logic and plug it in if the base package logic is not satisfactory.

Clicking **Cancel** causes the PA contract to become canceled. Cancellation should be used when you want to "logically delete" a PA contract because it shouldn't have been created.

The logic described above for breaking a payment arrangement is executed when a user cancels a payment arrangement; the only difference is that the PA contract is marked with a different characteristic type / value than when it is broken.

How To Set Up A Payment Arrangement

To set up a new payment arrangement:

- Open Payment Arrangement for Bills in add mode.
- Select the **Account** in question.
- Select the debt to be transferred to the payment arrangement (by turning on/off the checkboxes adjacent to the **Unpaid Bills**).
- Enter the number of **Installments** or the **Arrange Amount**.
- Define the **Division** and **Contract Type** of the new payment arrangement contract.
- Click the **Create** button.

How To Add Additional Debt To A Payment Arrangement

To transfer additional debt to an existing payment arrangement:

- Open Payment Arrangement for Bills in update mode.
- Select the existing payment arrangement in the search.
- Select the debt to be transferred to the payment arrangement (by turning on/off the checkboxes adjacent to the **Unpaid Bills** amounts).
- Enter the number of **Installments** or the **Arrange Amount**.
- Click the **Change** button.

> **Note: After transfer.** After transferring the funds to the payment arrangement contract, confirm the account's payoff and current balances make sense (the payoff balance is the total amount of debt that the account will eventually have to payoff, the current balance is how much they currently owe. The account's payoff and current balances are displayed in the page's second section. If you do not understand the difference between payoff balance and current balance, refer to Current Amount versus Payoff Amount.

How To Break A Payment Arrangement

You would break a payment arrangement when a customer doesn't make the agreed payments. Typically, your overdue processes will break a payment arrangement behind-the-scenes if a customer doesn't pay. Refer to the Breaking A Bill Oriented Payment Arrangement section for information about how the system monitors payment arrangements.

If you want to manually break a payment arrangement,

- Open Payment Arrangement for Bills in update mode.
- Select the existing payment arrangement in the search.

Copyright © 2024, Oracle and/or its affiliates.

- Click the **Break** button.

How To Cancel A Payment Arrangement

You would cancel a payment arrangement if you made a mistake and never intended to set up the payment arrangement in the first place. To cancel a payment arrangement

- Open Payment Arrangement for Bills in update mode.
- Select the existing payment arrangement in the search.
- Click the **Cancel** button.

Copyright © 2024, Oracle and/or its affiliates.

Chapter 18

Data Extractor

Topics:

- Data Extraction
- Modes of Data Extraction
- Prerequisites
- Creating the DTEXSCH Feature Configuration
- Data Extract Request Type
- Data Extract Request Dashboard

Oracle Revenue Management and Billing provides you with a facility to extract the system data in the **Comma Separated Values (CSV)** format. You can extract transactional data (such as, bills, payments, adjustments, and so on) as well as non-transactional data (i.e., master data, such as persons, accounts, adjustment types, bill segment types, and so on) from the system. You can extract the data in either batch or online mode. The extracted data can then be used by the downstream system for various purposes, such as accounting, financial analysis, payment processing, streamlining operations, reporting, and so on.

The **Data Extractor** module provides the following features:

- Supports data extraction from each table associated with the respective maintenance object.
- Facilitates full data extraction at the first instance and incremental data extraction in the subsequent instances.
- Provision to setup full or incremental data extraction in the subsequent instances.
- Separate CSV file created for each table from where the data is extracted.
- Facilitates the customers to configure the data extraction at required intervals.
- Enables you to define data masking rules for sensitive data.
- Supports conditional filters to extract selective data.

You can also use an OUAF feature named **Generalized Data Export (GDE)**. It facilitates you to extract the system data in the **JavaScript Object Notation (JSON)** format. For more information, see the **Generalized Data Export** section in the *Oracle Utilities Application Framework Administrative Guide*. There is one problem with GDE incremental data loading. It does not work as expected for the old legacy or Cobol entities (such as person, account, contract). To handle this issue, while configuring a data export control for a maintenance object for incremental data capture in the **Generalized Export Dashboard** screen, you must do the following:

- Remove the **F1-MO-REGCHN** algorithm attached to the **Audit** system event of the respective maintenance object.
- Attach the **C1-MO-REGCHN** algorithm to the **Post-Processing** system event of the respective business object.

Note: We recommend you to use the **Generalized Data Export (GDE)** feature while extracting the data created using the non-transactional business objects.

Data Extraction

The system enables you to extract the data from the system in the CSV file through a batch control named **C1-DTEX**. While extracting the data through the **C1-DTEX** batch, you need to specify the data extract request type using which you want to create the data extract request. The data extract request type indicates the business object whose data you would like to extract at the predefined location. The system then derives the data created using the business object from each table associated with the respective maintenance object. A separate CSV file is created for each table from where the data is extracted. Let us understand this with the help of an example.

While extracting the data for the **C1_BILL_BO** business object, the system will extract the data from each table associated with the **BILL** maintenance object and will create one CSV file for each table (i.e., CI_BILL, CI_BILL_CHAR, CI_BILL_EXCP, CI_BILL_MSG_PRM, CI_BILL_MSGS, CI_BILL_ROUTING, CI_BILL_SA) and one file to record the errors (if any).

For example, if you extract the bill information using the **DEXT** data extract request type where the business object is set to **C1_BILL_BO**, the system will create the following CSV files at the predefined location:

- DEXT_BILL_CI_BILL_07-01-2023_243.csv
- DEXT_BILL_CI_BILL_CHAR_07-01-2023_243.csv
- DEXT_BILL_CI_BILL_EXCP_07-01-2023_243.csv
- DEXT_BILL_CI_BILL_MSG_PRM_07-01-2023_243.csv
- DEXT_BILL_CI_BILL_MSGS_07-01-2023_243.csv
- DEXT_BILL_CI_BILL_ROUTING_07-01-2023_243.csv
- DEXT_BILL_CI_BILL_SA_07-01-2023_243.csv
- DEXT_BILL_ERROR_07-01-2023_243.csv

Here, the file naming convention used is as follows:

```
<Data_Extract_Request_Type>_<MO>_<Table_Name>_<Extraction_Date>_<Batch_Run_No>.csv
```

The system allows you to extract data created using a business object only when it is set in the algorithm attached to **Determine BO** system event of the respective maintenance object. For example, while creating a data extract request type for the **C1_BILL_BO** business object, you need to ensure that the **Business Object** parameter in the algorithm attached to the **Determine BO** system event of the **BILL** maintenance object is set to **C1_BILL_BO**. Otherwise, the data created using the **C1_BILL_BO** business object will not be extracted through the data extract request type.

You need to define a data extract request type for each business object whose transactional or non-transactional data you want to extract. On extracting the data, all or filtered data (based on the condition) is extracted in the CSV file. The CSV file includes all the columns from the table including two additional columns named **DT_EXTRACT_DTTM** and **RECORD_STATUS**. The **DT_EXTRACT_DTTM** column contains the date and time when the record is extracted from the table. On full data extraction, the system sets the **RECORD_STATUS** column for each record in the CSV file to **I** (i.e., Add). However, on incremental data extraction, the system sets the **RECORD_STATUS** column for each record in the CSV file to either **I** (i.e., Add) or **U** (i.e., Update) depending on whether the data is added or updated in the system since the last extraction.

Transactional Business Objects

A transactional business object is a business object using which high volume transactional data, such as bills, payments, adjustments, etc. is created in the system. Data created using such transactional business objects can only be extracted in the batch mode and not in the online mode. In the first batch run, the data is fully extracted from the tables irrespective of the data extract type (i.e., full or incremental) specified in the data extract request type. However, in the subsequent batch run, the data is extracted based on the data extract type (i.e., full or incremental) specified in the data extract request type.

The following table lists all the transactional business objects for which you can configure data extraction:

Copyright © 2024, Oracle and/or its affiliates.

Transactional Business Object	**Helps to maintain…**	**Comments**
C1-Bill	Bills	All the bills irrespective of their status are extracted.
C1-BILL_SEG_BO	Bill Segments	All the bill segments irrespective of their status are extracted.
C1-Adjustment	Adjustments	All the adjustments with the **Frozen**, **Canceled**, or **Error** status are extracted.
C1-PAY_BO	Payments and Payment Segments	All the payments with the **Frozen**, **Canceled**, or **Error** status are extracted.
C1-BILLCHARGE	Billable Charges	All the billable charges with the **Billable** or **Canceled** status are extracted.
C1-FT	Financial Transactions	All the financial transactions with the **Frozen** status are extracted.
C1-TXNTFMDTL	Transaction Details	All the transactions irrespective of their status are extracted.
C1-TxnHeader	Transaction Headers	All the transaction headers are extracted.
C1-MATCH_EVENT	Match Events	All the match events with the **Balanced** or **Canceled** status are extracted.
C1-DEPOSIT_CTL_BO	Deposit Controls	All the deposit controls with the **Balanced** status are extracted.
C1-NonCISPaymentEvent	Payment Events	All the payment events along with their payments are extracted.
C1-DelinquencyProcess	Delinquency Process	All the delinquency processes irrespective of their status are extracted.
C1-TNDR_CTL	Tender Controls	All the tender controls with the **Balanced** status are extracted.
C1-Benefits	Membership Benefits	All the membership benefits irrespective of their status are extracted.
C1-PriceCalculation	Price Calculations	All the price calculations are extracted.
C1_PERSON_BO	Persons	All the persons are extracted.
C1-AccountBO	Accounts	All the accounts are extracted.
C1_SA	Contracts	All the contracts irrespective of their status are extracted.
C1-CC	Customer Contacts	All the customer contacts irrespective of their status are extracted.

Note: The data extraction is filtered based on the filter condition (if any) specified in the data extract request type.

Copyright © 2024, Oracle and/or its affiliates.

Master Data Business Objects

A master data or non-transactional business object is a business object using which the functional data (such as, persons, accounts, price items, etc.) or configuration data (such as, algorithms, rate schedule, rate components, etc.) is created in the system. Data created using such non-transactional business objects can be extracted in the batch and online modes. Note that the online mode is supported only for non-transactional business objects that are User Interface (UI) based. In the first batch run, the data is fully extracted from the tables irrespective of the data extract type (i.e., full or incremental) specified in the data extract request type. However, subsequently, you can extract incremental data in the batch or online mode.

Some examples of non-transactional business object are:

- C1_RATECOMPONENT
- C1-AlgorithmBO

Alternatively, you can use the **Generalized Data Export (GDE)** feature to extract data created using the non-transactional business objects.

Modes of Data Extraction

The **Data Extractor** feature enables you to extract data in two different modes depending upon the type of the business object. The two different data extraction modes are:

- **Batch Mode** - The batch mode enables you to extract data from the ORMB application by executing or scheduling **C1-DTEX** batch. It enables you to extract full or incremental data created using a business object. It checks the data extract type specified in the data extract request type. If the data extract type is set to **Full**, the batch extracts the entire data from all the tables associated with the respective maintenance object. However, if the data extract type is set to **Incremental**, the batch extracts the delta (i.e., data which is added or updated since the last extraction) from all the tables associated with the respective maintenance object. On executing the **C1-DTEX** batch, the system extracts the data in multiple files and initially assigns the .part extension indicating that the data extraction is still in progress. Once the data extraction is completed, the extension changes to .csv confirming successful completion of the data extraction process.
- **Online Mode** - The online mode enables you to extract incremental data created using a business object. Note that the online mode is supported only for non-transactional business objects that are User Interface (UI) based. To enable the online mode data extraction, you need to attach an algorithm created using the **DTEXBOPSTALG** algorithm type to the **Post-Processing** system event of the respective non-transactional business object. In the online mode, the system extracts the data immediately when it is added or updated through the user interface. The data is extracted in the CSV file at the predefined location. Note that the deleted records are not extracted in the CSV file. For example, when you configure online mode data extraction for the **C1_PERSON_BO** business object, the system immediately extracts the person record that is added or updated in the CSV file. You can schedule a time slot for online mode data extraction by setting the **DTEXSCH** feature configuration. This feature configuration enables you to maintain a single file for each table with consolidated records added or updated during the day in the online mode instead of creating multiple files for each table throughout the day. For more information, see Creating the DTEXSCH Feature Configuration on page 869

If any error occurs while extracting data, the errors are stored in the following file, which is created along with the CSV files:

```
<Data_Extract_Request_Type>_<MO>_ERROR_<Extraction_Date>_<Batch_Run_No>.csv
```

The summary of the extracted files is maintained in the **CI_DATA_EX_REQ** table. The data created using the transactional business objects can only be extracted in the batch mode. However, the data created using the non-transactional business objects can be extracted in the batch mode as well as in the online mode. The system supports full data extraction only through the batch mode and not in the online mode. However, the system supports incremental data extraction in both online and batch modes.

Copyright © 2024, Oracle and/or its affiliates.

Prerequisites

To setup the data extractor feature, you need to do the following:

- Define the required data extract request types in the system
- Attach an algorithm to the **Determine BO** system event of the maintenance object whose business object data you want to extract and ensure that the algorithm parameter is set to the respective business object
- Create the **DTEXSCH** feature configuration, if required, for the online mode data extraction
- Set the batch control type of the **Data Extract (C1-DTEX)** batch to **Timed** and define the following attributes:
  - Time Interval
  - Timer Active
  - User ID
  - Batch Language
  - Email Address

  For more information about this batch, see *Oracle Revenue Management and Billing Batch Guide*.

Creating the DTEXSCH Feature Configuration

The **DTEXSCH** feature configuration enables you to maintain a single file for each table with consolidated records added or updated during the day in the online mode instead of creating multiple files for each table throughout the day.

Procedure

To create the **DTEXSCH** feature configuration:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **F** and then click **Feature Configuration**.
   A sub-menu appears.
3. Click the **Add** option from the **Feature Configuration** sub-menu.
   The **Feature Configuration** screen appears.
4. Enter `DTEXSCH` in the **Feature Name** field.
5. Select the **Data Extraction Schedule** option from the **Feature Type** list.
6. Enter the appropriate description in the respective field.
7. Set the following option type in the feature configuration:

Option Type	Description	Mandatory (Yes or No)
Time Slot	Used to indicate the time slot defined for online mode data extraction. The system will then create a CSV file for each table associated with the maintenance object. The system will extract all the records which are added or updated between the given time slot.	No
	Note: You must specify the time slot in the HH:MI HH:MI format using 24 Hours base. For example, 00:00 23:59. This means that the system will extract all the records which are added or updated between 12:00 am to 11:59 pm at once in the online mode.	

8. Click the **Save** button in the **Page Title** area.

Copyright © 2024, Oracle and/or its affiliates.

The feature configuration is defined.

Data Extract Request Type

Oracle Revenue Management and Billing allows you to define a data extract request type using which you can create a data extract request. It is the data extract request type which helps the system to determine the following:

- Data created using the business object should be extracted through a data extract request
- Whether full or incremental data should be extracted through a data extract request
- Whether sensitive information, such person information, account details, etc. should be masked to prevent unauthorized access
- Whether all or filtered data should be extracted during full or incremental extraction
- Directory or bucket where the data should be extracted in the **Comma Separated Values (CSV)** format

The **Data Extract Request Type** screen allows you to search and view the details of a data extract request type. It also allows you to define, edit, and delete a data extract request type. It contains the following zones:

- Search Data Extract Request Type on page 870
- Data Extract Request Type on page 871

Search Data Extract Request Type

The **Search Data Extract Request Type** zone allows you to search for a data extract request type using various search criteria. It contains the following two sections:

- **Search Criteria** - The **Search Criteria** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Data Extract Code	Used to search for a particular data extract request type.	No
Description	Used to search for data extract request types with a particular description.	No

- **Search Results** - On clicking the **Search** button, the search results appear based on the specified search criteria. The **Search Results** section contains the following columns:

Column Name	Column Description
Data Extract Code	Displays the data extract request type.
Description	Displays the description of the data extract request type.
Business Object	Indicates the business object for which the data extract request type is created in the system.
Edit	On clicking the **Edit** (✎) icon, the **Data Extract Request Type** screen appears where you can edit the details of the data extract request type.
Delete	On clicking the **Delete** (🗑) icon, you can delete the data extract request type. **Note:** You can delete a data extract request type only when a data extract request is not yet created using the data extract request type.

On clicking the **Broadcast** (📡) icon corresponding to a data extract request type, the **Data Extract Request Type** zone appears with the details of the respective data extract request type.

Copyright © 2024, Oracle and/or its affiliates.

Related Topics

For more information on...	See...
How to search for a data extract request type	Searching for a Data Extract Request Type on page 871
How to edit a data extract request type	Editing a Data Extract Request Type on page 876
How to delete a data extract request type	Deleting a Data Extract Request Type on page 880
How to view the details of a data extract request type	Viewing the Data Extract Request Type Details on page 880

Data Extract Request Type

The **Data Extract Request Type** zone displays the details of the data extract request type. It contains the following fields:

Field Name	Field Description
Data Extract Code	Displays the data extract request type.
Description	Displays the description of the data extract request type.
Business Object	Indicates the business object for which the data extract request type is created in the system. **Note:** It has a link. On clicking the link, the **Business Object** screen appears where you can view the details of the respective business object.
Data Extract Type	Indicates how the data created using the business object should be extracted. The valid values are: • Full • Incremental
Data Masking Rules	Displays the masking rules using which sensitive information should be hidden from the unauthorized users.
Filter Condition	Displays the condition using which the data is filtered during full or incremental extraction.
Data Extract Directory Path	Displays the directory or bucket where the data is extracted in the **Comma Separated Values (CSV)** format.

By default, the **Data Extract Request Type** zone does not appear in the **Data Extract Request Type** screen. It appears when you click the **Broadcast** (⦿) icon corresponding to the data extract request type in the **Search Data Extract Request Type** zone.

Related Topics

For more information on...	See...
Data Extract Request Type screen	Data Extract Request Type on page 870
Data Extract Request Type zone	Search Data Extract Request Type on page 870

Searching for a Data Extract Request Type

Procedure

Copyright © 2024, Oracle and/or its affiliates.

To search for a data extract request type:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **D** and then click **Data Extract Request Type**.
   A sub-menu appears.
3. Click the **Search** option from the **Data Extract Request Type** sub-menu.
   The **Data Extract Request Type** screen appears.
4. Enter the search criteria in the **Search Data Extract Request Type** zone.

   Note: ORMB search engine supports wildcard search, where you can substitute the percentage (%) symbol as a stand in for any word or letter in a search criteria. You can use the '%' wildcard character in all input fields except the date and ID fields. The '%' wildcard character is suffixed automatically at the end of the partial search criteria. Therefore, you may or may not specify the wildcard character at the end of the partial search criteria. However, you have to prefix the wildcard character manually wherever required.
5. Click **Search**.
   A list of data extract request types that meet the search criteria appears in the **Search Results** section.

Related Topics

For more information on...	See...
Data Extract Request Type screen	Data Extract Request Type on page 870
Search Data Extract Request Type zone	Search Data Extract Request Type on page 870
Data Extract Request Type zone	Data Extract Request Type on page 871

Defining a Data Extract Request Type

Prerequisites

To define a data extract request type, you should have:

- Business objects defined in the application

Procedure

To define a data extract request type:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **D** and then click **Data Extract Request Type**.
   A sub-menu appears.
3. Click the **Add** option from the **Data Extract Request Type** sub-menu.

   The **Data Extract Request Type** screen appears. It contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Data Extract Code	Used to specify the data extract request type.	Yes
Description	Used to specify the description for the data extract request type.	No
Business Object	Used to indicate the business object for which you want to create a data extract request type. The system will then extract the data created using the specified business object through the data extract request type.	Yes

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: The **Search** (🔍) icon appears corresponding to the **Business Object** field. On clicking the **Search** icon, the **Business Object Search** window appears. On selecting a business object, the description of the business object appears corresponding to the **Business Object** field. Here, you must select a business object only when it is set in the algorithm attached to **Determine BO** system event of the respective maintenance object. For example, while creating a data extract request type for the **C1_PERSON_BO** business object, you need to ensure that the **Business Object** parameter in the algorithm attached to the **Determine BO** system event of the respective maintenance object is set to **C1_PERSON_BO**. Otherwise, the data created using the business object will not be extracted through the data extract request type.	
Data Extract Type	Used to indicate how you want to extract the data created using the business object. The valid values are: • **Full** - Used when you want to extract the entire data from all the tables associated with the respective maintenance object of the business object. The system supports full data extraction only through a batch process (i.e., when the **C1-DTEX** batch is executed) and not through the online mode. • **Incremental** - Used when you want to extract the delta (i.e., data which is added or updated since the last extraction) from all the tables associated with the respective maintenance object of the business object. The system supports incremental data extraction in the online and batch modes. The incremental data extraction in the batch mode is supported only for the transactional business objects. However, the incremental data extraction in the online mode is supported only for those business objects which are User Interface (UI) based.	Yes

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: While extracting the data for the first time, you need to extract it through a batch process (i.e., when the **C1-DTEX** batch is executed). However, while extracting the data subsequently, you can extract it either in the online or batch mode. On full data extraction, the system sets the **RECORD_STATUS** column for each record in the CSV file to **I** (i.e., Add). However, on incremental data extraction, the system sets the **RECORD_STATUS** column for each record in the CSV file to either **I** (i.e., Add) or **U** (i.e., Update) depending on whether the data is added or updated in the system.	

Data Masking Rules	Used to specify masking rules using which sensitive information should be hidden from the unauthorized users. You can specify the data masking rule for a column in the following format:
	`column_name:number_of_characters:F/L:masking_character`
	Where,
	• **column_name** - Here, you need to specify the column name whose data you want to mask. Note that you can only specify the column of a primary table of the respective maintenance object. For example, PHONE_TYPE_CD.
	• **number_of_characters** - Here, you need to specify the number of characters that you want to mask. For example, 7.
	• **F/L** - Here you need to indicate whether the masking should be done from the beginning or from the end. The valid values are:
	  • **F** - Used when you want to mask the specified number of characters starting from the beginning (i.e., from the first character).
	  • **L** - Used when you want to mask the specified number of characters starting from the end (i.e., from the last character).
	• **masking_character** - Here you need to specify the character using which you want to mask the original characters. For example, X.
	Let us assume that you specify the following data masking rule for the PHONE_TYPE_CD column:
	`PHONE_TYPE_CD:7:F:X`
	Then, on extracting the data, the system will mask the phone number and accordingly display XXXXXXX119 corresponding the record in the CSV file.
		No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: You can specify data masking rules for multiple columns separated by the **Semi Colon** (;) symbol. For example, EMAILID:15:F:X;PHONE_TYPE_CD:7:F:X.	
Filter Condition	Used to specify the condition using which you want to filter the data during full or incremental extraction. The system then automatically generates the WHERE clause using the filter condition while extracting the data. You must specify the condition in the following format: `column_name='column_value'` For example, `COUNTRY='USA'`. **Note:** You can only specify the column of a primary table of the respective maintenance object in the filter condition. The column value should be specified in single quotation marks ('). While defining the filter condition, you can specify the following: • SQL logical operator, such as AND, OR, NOT, LIKE, etc. • Multiple column and column value combinations based on which you want to filter the data extraction For example, `COUNTRY='USA' AND CITY='San Francisco'`.	No
Data Extract Directory Path	Used to specify the directory or bucket where you want to extract the data in the **Comma Separated Values (CSV)** format. The specified directory path is referred while extracting the data in either online or batch mode. If the application environment is an On Premise or Cloud Service environment, you can specify the directory name along with the following relative path: • **@SHARED_DIR** – Used when you want to store the CSV file in the shared directory on the server. **Note:** The shared directory variable (i.e., **SHARED_DIR**) is case-sensitive. For example, you can set the value to `@SHARED_DIR/extract`. However, if the application environment is a Cloud Service Premium Edition environment, you must specify the following information in the `file-storage://<File Storage Extendable`	Yes

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	`Lookup Value>/<Object Storage Bucket>` format: • The file storage extendable lookup value from where the cloud storage compartment and connection details are retrieved while extracting the data on OCI • The bucket or container where you want the system to extract the data in the CSV format **Note:** You must ensure that you have write permissions on the specified directory or bucket.	

Note: Alternatively, you can access this screen by clicking the **Add** button in the **Page Title** area of the **Data Extract Request Type** screen.

4. Enter the required details in the **Data Extract Request Type** screen.
5. Click **Save**.
   The data extract request type is created.

Related Topics

For more information on...	See...
Data Extract Request Type screen	Data Extract Request Type on page 870

Editing a Data Extract Request Type

Procedure

To edit a data extract request type:

1. Search for the data extract request type in the **Data Extract Request Type** screen.
2. In the **Search Results** section, click the **Edit** icon in the **Edit** column corresponding to the data extract request type whose details you want to edit.

   The **Data Extract Request Type** screen appears. It contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Data Extract Code	Displays the data extract request type.	Not applicable
Description	Used to specify the description for the data extract request type.	No
Business Object	Used to indicate the business object for which you want to create a data extract request type. The system will then extract the data created using the specified business object through the data extract request type.	Yes

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: The **Search** (🔍) icon appears corresponding to the **Business Object** field. On clicking the **Search** icon, the **Business Object Search** window appears. On selecting a business object, the description of the business object appears corresponding to the **Business Object** field. Here, you must select a business object only when it is set in the algorithm attached to **Determine BO** system event of the respective maintenance object. For example, while creating a data extract request type for the **C1_PERSON_BO** business object, you need to ensure that the **Business Object** parameter in the algorithm attached to the **Determine BO** system event of the respective maintenance object is set to **C1_PERSON_BO**. Otherwise, the data created using the business object will not be extracted through the data extract request type.	
Data Extract Type	Indicates how the data created using the business object should be extracted. The valid values are: • **Full** - Extracts the entire data from all the tables associated with the respective maintenance object of the business object. The system supports full data extraction only through a batch process (i.e., when the **C1-DTEX** batch is executed) and not through the online mode. • **Incremental** - Extracts the delta (i.e., data which is added or updated since the last extraction) from all the tables associated with the respective maintenance object of the business object. The system supports incremental data extraction in the online and batch modes. The incremental data extraction in the batch mode is supported only for the transactional business objects. However, the incremental data extraction in the online mode is supported only for those business objects which are User Interface (UI) based.	Not applicable

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: While extracting the data for the first time, you need to extract it through a batch process (i.e., when the **C1-DTEX** batch is executed). However, while extracting the data subsequently, you can extract it either in the online or batch mode. On full data extraction, the system sets the **RECORD_STATUS** column for each record in the CSV file to **I** (i.e., Add). However, on incremental data extraction, the system sets the **RECORD_STATUS** column for each record in the CSV file to either **I** (i.e., Add) or **U** (i.e., Update) depending on whether the data is added or updated in the system.	

Data Masking Rules	Used to specify masking rules using which sensitive information should be hidden from the unauthorized users. You can specify the data masking rule for a column in the following format:
	`column_name:number_of_characters:F/L:masking_character`
	Where,
	• **column_name** - Here, you need to specify the column name whose data you want to mask. Note that you can only specify the column of a primary table of the respective maintenance object. For example, PHONE_TYPE_CD.
	• **number_of_characters** - Here, you need to specify the number of characters that you want to mask. For example, 7.
	• **F/L** - Here you need to indicate whether the masking should be done from the beginning or from the end. The valid values are:
	  • **F** - Used when you want to mask the specified number of characters starting from the beginning (i.e., from the first character).
	  • **L** - Used when you want to mask the specified number of characters starting from the end (i.e., from the last character).
	• **masking_character** - Here you need to specify the character using which you want to mask the original characters. For example, X.
	Let us assume that you specify the following data masking rule for the PHONE_TYPE_CD column:
	`PHONE_TYPE_CD:7:F:X`
	Then, on extracting the data, the system will mask the phone number and accordingly display XXXXXXX119 corresponding the record in the CSV file.
		No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: You can specify data masking rules for multiple columns separated by the **Semi Colon** (;) symbol. For example, EMAILID:15:F:X;PHONE_TYPE_CD:7:F:X.	
Filter Condition	Used to specify the condition using which you want to filter the data during full or incremental extraction. The system then automatically generates the WHERE clause using the filter condition while extracting the data. You must specify the condition in the following format: `column_name='column_value'` For example, `COUNTRY='USA'`. **Note:** You can only specify the column of a primary table of the respective maintenance object in the filter condition. The column value should be specified in single quotation marks ('). While defining the filter condition, you can specify the following: • SQL logical operator, such as AND, OR, NOT, LIKE, etc. • Multiple column and column value combinations based on which you want to filter the data extraction For example, `COUNTRY='USA' AND CITY='San Francisco'`.	No
Data Extract Directory Path	Used to specify the directory or bucket where you want to extract the data in the **Comma Separated Values (CSV)** format. The specified directory path is referred while extracting the data in either online or batch mode. If the application environment is an On Premise or Cloud Service environment, you can specify the directory name along with the following relative path: • **@SHARED_DIR** – Used when you want to store the CSV file in the shared directory on the server. **Note:** The shared directory variable (i.e., **SHARED_DIR**) is case-sensitive. For example, you can set the value to `@SHARED_DIR/extract`. However, if the application environment is a Cloud Service Premium Edition environment, you must specify the following information in the `file-storage://<File Storage Extendable`	Yes

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	`Lookup Value>/<Object Storage Bucket>` format: • The file storage extendable lookup value from where the cloud storage compartment and connection details are retrieved while extracting the data on OCI • The bucket or container where you want the system to extract the data in the CSV format **Note:** You must ensure that you have write permissions on the specified directory or bucket.	

3. Modify the required details in the **Data Extract Request Type** screen.
4. Click **Save**.
   The changes made to the data extract request type are saved.

Related Topics

For more information on...	See...
How to search for a data extract request type	Searching for a Data Extract Request Type on page 871
Data Extract Request Type screen	Data Extract Request Type on page 870
Data Extract Request Type zone	Data Extract Request Type on page 871

Deleting a Data Extract Request Type

Procedure

To delete a data extract request type:

1. Search for the data extract request type in the **Data Extract Request Type** screen.
2. In the **Search Results** section, click the **Delete** (🗑) icon in the **Delete** column corresponding to the data extract request type that you want to delete.

   A message appears confirming whether you want to delete the data extract request type.

   Note: You can delete a data extract request type only when a data extract request is not yet created using the data extract request type.

3. Click **OK**.
   The data extract request type is deleted.

Related Topics

For more information on...	See...
How to search for a data extract request type	Searching for a Data Extract Request Type on page 871
Data Extract Request Type screen	Data Extract Request Type on page 870

Viewing the Data Extract Request Type Details

Procedure

Copyright © 2024, Oracle and/or its affiliates.

To view the details of a data extract request type:

1. Search for the data extract request type in the **Data Extract Request Type** screen.
2. In the **Search Results** section, click the **Broadcast** ( ) icon corresponding to the data extract request type whose details you want to view.
   The **Data Extract Request Type** zone appears.
3. View the details of the data extract request type in the **Data Extract Request Type** zone.

Related Topics

For more information on...	See...
How to search for a data extract request type	Searching for a Data Extract Request Type on page 871
Data Extract Request Type screen	Data Extract Request Type on page 870
Data Extract Request Type Display zone	Data Extract Request Type on page 871

Data Extract Request Dashboard

The **Data Extract Request Dashboard** screen allows you to view the status of various data extract requests initiated through a batch process. It contains the following zone:

- Search Data Extract Request on page 881

Search Data Extract Request

The **Search Data Extract Request** zone allows you to search for a data extract request using various search criteria. It contains the following two sections:

- **Search Criteria** - The **Search Criteria** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Data Extract Code	Used to search for data extract requests which are created through a particular data extract request type. **Note:** You must specify a data extract request type which is already defined in the system.	No
File Name	Used to search for a data extract request through which a particular CSV file is created.	No
Extract Date	Used to search for data extract requests which are created on a particular date. **Note:** You can either manually specify the date or select it using the **Date Picker** ( ) icon corresponding to the field.	No
Status Code	Used to search for data extract requests with a particular status. The valid values are: • SUCCESS • INPROGRESS	No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	• ERROR	

- **Search Results** - On clicking the **Search** button, the search results appear based on the specified search criteria. The **Search Results** section contains the following columns:

Column Name	Column Description
Data Extract Code	Indicates the data extract request type using which the data extract request is created.
File Name	Indicates the CSV file that is extracted through the data extract request type.
Extract Date	Displays the date when the CSV file is extracted.
Failed Records	Indicates the number of records which could not be successfully extracted from the system.
Success Records	Indicates the number of records which are successfully extracted from the system.
Status	Indicates the status of the data extract request. The valid values are: • SUCCESS • INPROGRESS • ERROR

Related Topics

For more information on...	See...
How to search for a data extract request	Searching for a Data Extract Request on page 882

Searching for a Data Extract Request

Procedure

To search for a data extract request:

1. Click the **Menu** link in the **Application** toolbar.
   A list appears.
2. From the **Main** menu, select **Integration** and then click **Data Extract Request Dashboard**.
   The **Data Extract Request Dashboard** screen appears.
3. Enter the search criteria in the **Search Data Extract Request** zone.

   Note: ORMB search engine supports wildcard search, where you can substitute the percentage (%) symbol as a stand in for any word or letter in a search criteria. You can use the '%' wildcard character in all input fields except the date and ID fields. The '%' wildcard character is suffixed automatically at the end of the partial search criteria. Therefore, you may or may not specify the wildcard character at the end of the partial search criteria. However, you have to prefix the wildcard character manually wherever required.

4. Click **Search**.
   A list of data extract requests that meet the search criteria appears in the **Search Results** section.

Related Topics

For more information on...	See...
Data Extract Request Dashboard screen	Data Extract Request Dashboard on page 881

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
Search Data Extract Request zone	Search Data Extract Request on page 881

Copyright © 2024, Oracle and/or its affiliates.

Chapter 19

File Upload Interface

Topics:

- File Upload Process
- Prerequisites
- Configuring Data Transformation in a File Request Types
- Configuring the Approval Workflow Process for File Upload Interface
- Configuring Rules Engine for Polling Files on SFTP Server
- Setting the C1-FLUPLD Feature Configuration
- Setting the File Upload Interface Master Configuration
- Setting the Email Sender Master Configuration
- Setting the File Management System Configuration
- File Management System
- File Request Type
- File Upload Dashboard
- File Record Details

The Oracle Revenue Management and Billing **File Upload Interface** allows users to upload files into a staging area and map file records to Oracle Revenue Management and Billing services. Each record in the file is processed individually by invoking the relevant services.

The application provides a **File Upload Interface Configuration** that sets the rules and parameters for uploading and transforming files. Users can create a **File Request Type**, which serves as a template to guide how the file or records are uploaded and processed.

Each file is processed based on its file request type, ensuring consistency and accuracy. Records from the file are staged one by one using the File Upload and Transform **C1-FTRAN** batch process, which helps the system understand how to handle and process the uploaded files effectively.

File Upload Process

The file upload process takes place either through a batch which allows to upload an XML file having record(s) in compliance with ORMB or by transforming the uploaded file into a specified customer format. Following methods explain the file upload process in detail.

- Upload and Transforming a File on page 888
- Processing a File Request on page 893
- Archiving a File and/or File Records on page 897
- Polling Files on SFTP Server on page 897

Decrypting a File

Oracle Revenue Management and Billing provides you the ability to decrypt an encrypted file during the file upload process. To decrypt a file during the upload process you should have:

- **File Decryption Required** option selected in the **Master Configuration** screen for the file upload interface
- **File Decryption Required** option selected in the **File Request Type** screen using which the file is uploaded in the system
- The **Cipher Type** selected in the **Master Configuration** screen for the file upload interface for which you want to decrypt the file
- The **C1-FTRAN** batch executed in the application

The system supports decryption for the following cipher types:

- **AES – Advanced Encryption Standards** - Advanced Encryption Standards(AES) is a symmetric-key encryption which uses only one secret key to encrypt and decrypt information. The secret key is captured while defining the master configuration for the file upload interface. The algorithm used to encrypt or decrypt a file using the AES cipher type is **AES/CBC/PKCS5Padding**.

  To decrypt a file using the **Advanced Encryption Standards** cipher type, you need to do the following:

  - Save the master configuration by selecting the **Cipher Type** as **Advanced Encryption Standards**.
  - Ensure that the secret key set to decrypt the file is not more than 16 bytes.

    Tip: You can save the configuration with a 16 byte key like `1234567812345678123456`.

  - Execute the **C1-FTRAN** batch.
- **DES – Data Encryption Standard** - Data Encryption Standard (DES) is a symmetric-key encryption which uses only one secret key to encrypt and decrypt information. Therefore, both the sender and the receiver must know and use the same secret key. The secret key is captured while defining the master configuration for the file upload interface. The algorithm used to encrypt or decrypt a file using the DES cipher type is **DES/CBC/PKCS5Padding**.

  To decrypt a file using the **Data Encryption Standard** cipher type, you need to:

  - Save the master configuration by selecting the **Cipher Type** as **Data Encryption Standard**.
  - Ensure that the secret key set to decrypt the file is not more than 8 bytes.

    Tip: You can save the configuration with a 16 byte key like `12345678123`.

  - Execute the **C1-FTRAN** batch.
- **PGP- Pretty Good Privacy** - Pretty Good Privacy (PGP) is a hybrid cryptosystem that combines symmetric-key encryption with public-key (asymmetric) encryption.

  To decrypt a file using the **Pretty Good Privacy** cipher type, you need to:

  - Generate the private and public keys using the **Kleopatra** tool and save the keys with **.asc** extension.

Copyright © 2024, Oracle and/or its affiliates.

- Create a folder named **PGP_KEYS** inside the `SHARED_DIR` and save the private and public keys in the **PGP_KEYS** folder.
- Add the values of the file path in the **C1-FLUPLD** feature configuration.
- In the **Master Configuration** screen, select the **Cipher Type** as **Pretty Good Privacy**, upload the generated private key file path in the **Upload Private Key** field, and enter the key in the **Pass Phrase** field which is generated using the **Kleopatra** tool.
- Save the master configuration.
- Encrypt the target file using the **Kleopatra** tool.
- Place the encrypted file in the `SHARED DIR`.
- Execute the **C1-FTRAN** batch.

- **RSA with AES** - RSA and AES are combined to build secure systems for protecting data and communications. In this hybrid approach, AES handles the actual data encryption due to its speed and efficiency, while RSA encrypts the AES key for secure key exchange. As a result, even if encrypted data is intercepted, it remains inaccessible without the RSA-encrypted AES key. This layered approach ensures that even if an attacker captures the AES-encrypted data, they won't be able to decrypt it without access to the AES key, which is protected by RSA encryption. RSA encryption and decryption use the **OAEPWithSHA-256AndMGF1Padding** algorithm.

  RSA-AES decryption can be performed in three different scenarios:

  - The RSA-encrypted AES key is stored in the `keystore` folder, while the AES-encrypted data file is located in the `SHARED_DIR` folder.
  - Both the RSA-encrypted AES key file and the AES-encrypted data file are stored in the `SHARED_DIR` folder.
  - A single file in the `SHARED_DIR` folder contains both the RSA-encrypted AES key and the AES-encrypted data.

  To decrypt a file using the **RSA with AES** cipher type, you need to do the following:

  - Place the data file in the project folder.
  - Generate RSA key pair by running the `genkey "XXX"` command. This will generate the following two files:
    - `XXX.key` - RSA private key
    - `XXX.pub` - RSA public key
  - Encrypt the data file by using the `enc xxx.pub filename.csv` command (for example: `enc fileBase.pub BILL_RT_TYPE_8004.csv`). This will generate the following files:
    - `filename.csv.enc` - This file contains both the encrypted AES key and AES-encrypted data
    - `AES_KEY.key` - This file contains the RSA-encrypted AES key.
    - `ENCRYPTED_DATA.csv` - This file contains the AES-encrypted version of the original file.
  - Select the **Cipher Type** as **RSA with AES** in the master configuration, enter the RSA private key in the **Private Key** field, and save the master configuration.
  - Restart the thread pool to apply changes.
  - Execute the **C1-FTRAN** batch.

  > **Note:** To test the decryption manually, run the `dec xxx.key filename.csv.enc` command using Java class and verify the decrypted data within the application.

Related Topics

For more information on...	See...
How to set the file upload interface master configuration	Setting the File Upload Interface Master Configuration on page 946
How to define a file request type	Defining a File Request Type on page 966
How to set the**C1-FLUPLD** feature configuration	Setting the C1-FLUPLD Feature Configuration on page 942

Copyright © 2024, Oracle and/or its affiliates.

Upload and Transforming a File

To upload a file in the ORMB application, you should use the **File Transformation and Upload (C1-FTRAN)** batch which uploads either an XML file that contains record(s) in compliance with the ORMB service schema or simply transforms and uploads the file in customer format, either in **XML** or **CSV** or **PSV** or **JSON** format. The **C1-FTRAN** batch uses the Record Transformation Algorithm to transform the file records into ORMB conform service schema and upload the same in the ORMB system. The uploaded file details are logged with the **FILE_ID** field in the **CI_FILE_REQUEST** table with either **Complete** or **Pending** status.

Note:

- File Record Transformation is only eligible for **XML** or **CSV** or **PSV** or **JSON** or **Fixed Formats**.
- The file will only be uploaded with **Pending** status when the all of the records in the file are not uploaded due to failure, else the file will be uploaded with **Complete** status.
- Every file will contain records for only a single File Request Type.

The **C1-FTRAN** batch is required to read the file(s) on the SFTP server at the given file path (passed as the soft parameter). If a **File Validation Algorithm** is mapped to the corresponding file request type then header, footer, and checksum validation can be performed. This batch reads one file at a time and adds a token on all of the records of the file. Once all the file records are read with a token, you must then create a **ThreadWorkUnit** for every single token, i.e. Number of records equal to the Number of Work-units. Before logging the file details and creating a **ThreadWorkUnit**, the file transformation batch validates the following conditions,

Condition	**Result**
If the File Path is invalid	Batch is terminated with **ERROR** status without logging any file details in the **CI_FILE_REQUEST** table.
If the File Path of Error Log is invalid	Batch is terminated with an **ERROR** status.
If the File does not exists in the path	The **C1-FTRAN** batch is terminated with an **ERROR** status without logging any file details within the **CI_FILE_REQUEST** table.
If the File name Extension does not matches with the one provided in Request Type Configuration	Batch will be terminated with **ERROR** status without logging any file details in the **CI_FILE_REQUEST** table.
If the file size exceeds the limit of 2GB	Batch is terminated with **COMPLETE** status. The file details are logged with an **ERROR** status and in the **CI_FILE_REQUEST** table.
If the File Request Type is incorrect	Batch is terminated with an **Error** status.
If encrypted files are uploaded, File Decryption process gets executed on those files. If File Decryption process fails	Batch is terminated with a **COMPLETE** status. The file and error details are logged with an **ERROR** status and in the**CI_FILE_REQUEST** table.
If the File Header (first line in the file) does not exist	Batch is terminated with a **COMPLETE** status. The file and error details are logged with an **ERROR** status and in the**CI_FILE_REQUEST** table.
If the File Footer (last line in the file) does not exist	Batch is terminated with a **COMPLETE** status. The file and error details are logged with an **ERROR** status and in the**CI_FILE_REQUEST** table.
If the File Header and File Footer are excluded, and the number of records do not match with number of lines in the file	Batch is terminated with a **COMPLETE** status. The file and error details are logged with an **ERROR** status and in the**CI_FILE_REQUEST** table.

Copyright © 2024, Oracle and/or its affiliates.

Condition	Result
If the uploaded file already exists in the ORMB system	Batch will be terminated with an **ERROR** status. The file and error details are logged with a **REJECT** status and in the**CI_FILE_REQUEST** table.
If the checksum validation (for example, search for **<FILENAME>.checksum** parameter to retrieve the checksum value) fails	Batch will be terminated with an **ERROR** status. The file and error details are logged with a **REJECT** status and in the**CI_FILE_REQUEST** table.

A list of **ThreadWorkUnit** is created by tokenizing the number of records in the corresponding file. Every single token is a single file record with unique **REQUEST_ID** identifier in **CI_FILE_REQUEST_DETAIL** table with **PENDING** status. All the files on SFTP server having same file extension corresponding to that configured for the given file record type will be read and uploaded in ORMB system. There are two fields to capture its corresponding XML (Payload) record data. Either of this field will be populated.

- **REQUEST** with **CLOB** DataType - This field is populated when the record with File Request Type has the configuration of **File Size Greater than 32KB**as **TRUE**.
- **BO_DATA_AREA** with **VARCHAR2(32000)** DataType - This will be populated when, request with File Request Type having configuration of **Record Size Greater than 32KB** as **FALSE**.

This handling is for better performance while uploading and processing file request data.

Note:

- File that already exists in ORMB system will be ignored.
- Java exceptions/Unknown errors will not be handled.

This batch supports both single-thread and multi-threaded execution using **File Atomicity** flag on **File Request Type**. Using the **errorLogFilePath (Execute batch using File Chunks)** soft-parameter this batch supports two ways for creating **ThreadWorkUnits** that are required for batch execution.

- Fetch and create **ThreadWorkUnits** for all these number of requests in **PENDING** or **RETRY** status for the given **File Request Type** from the **CI_FILE_REQUEST_DETAIL** table.

Field Name	Field Description
File ID	Used to specify a unique identifier for a file.
File Name	Used to specify the request object that is to be processed.
Request Type	Used to specify the file request type of the request object to be processed.
Request Object	Used to specify the payload / dataline in the application.
File Retry	Used to specify the value for the file retry field to skip the processed record in case the batch has been aborted.
Duplicate	Used to specify the value for the duplicate field to skip processing the request, if there are multiple records with the same identifier in a same file.
ID	Used to specify the value for the ID, specific to payment stage upload and transaction business services.
XPath	Used to specify the value for the XPath, specific to payment stage upload and transaction business services file request transformation.
Field Name	Used to specify the field name specific to payment stage upload and transaction business services file request transformation.

- Fetch and create **ThreadWorkUnits** for all these number of request(s) in **PENDING** or **RETRY** status for the given **File Request Type** from **CI_FILE_REQUEST_DETAIL** table.

Copyright © 2024, Oracle and/or its affiliates.

- Number of files are created in proportion to the number of threads, for example, if the number of threads are 100, then 100 files are created.
- These files will be created at the same file path on the SFTP server from where the uploaded files are being read.
- All these newly created files will have approximately equal number of requests i.e. if there are 1000 requests, then, each file will have 10 requests or records.
- ThreadWorkUnit will have supplementary data only for,

Field Name	Field Description
ID	Used to specify the value for the ID, specific to payment stage upload and transaction business services.
XPath	Used to specify the value for the XPath, specific to payment stage upload and transaction business services file request transformation.
Field Name	Used to specify the field name specific to payment stage upload and transaction business services file request transformation.

Note:

These new temporary files are deleted after all the requests in the file are uploaded in the ORMB system.

File Upload Interface Master Configuration

Attribute Name	Attribute Description
Validate Checksum	Checksum validation is performed before reading and staging the file details in the ORMB system.
Audit Log Required	Individual file request status (PENDING / PROCESSED / ERROR / SKIPPED / INPROGRESS / RETRY / IGNORE / RETRYLIMITEXCEED) transition will be logged in ORMB system
Archive File	In-case there is no file validation failure, file is moved to **Archive File Location** else file is moved to **Archive Error File** location.
File Encryption	In-case if the file encryption returns true, then the batch retrieves the encrypted file on SFTP server and decrypts the file using the **File Decryption** algorithm.

File Request Type Configuration

File Request Type (Attribute Name)	Attribute Description
File Transformation Required	• If the file transformation required attribute is true, the file request transformation is done using **Request Transformation Algorithm** i.e. (request in either CSV or PSV or XML or JSON to ORMB conform service schema). • If the attribute file transformation required is false, the payload that is provided within a file is used to invoke the corresponding service.
File Transformation Algorithm	• The file transformation algorithm is used for transforming the file request into an XML request, which is in compliance with the ORMB service schema. • A sample transformation algorithm **C1-FRTA** is provided. If you want to use this algorithm, then provide the **Map Field XPath** for every configured field, with **File Segment Type** as **Field Detail** in the **Transformation Details** section on the **File Request Type** configuration UI.

Copyright © 2024, Oracle and/or its affiliates.

File Request Type (Attribute Name)	**Attribute Description**
File Atomicity	• If the file atomicity attribute is true, then the **C1-FTRAN** batch is executed in a single thread mode, i.e. for **All or None Transactions**. • If the file atomicity attribute is false, then the **C1-FTRAN** batch is executed in multi-threaded mode. Multi-threading is based on number of requests within the file(s). For the multi-threaded batch we use the standard commit strategy.
File Upload and Process	• If the file upload and process attribute is true in the single-threaded mode, then for a single record transaction failure every other record transaction performed will be roll-backed. The file status will be marked as **Error** with its failure updates placed in **CI_FILE_REQUEST** table. All the corresponding file requests will be updated with an **Error** status in the **CI_FILE_REQUEST_DETAIL** table. • If the file upload and process attribute is true in multi-threaded batch, then for a single record transaction failure only that record transaction will be roll-backed. File status will be marked as **Complete** with its failure updates placed in **CI_FILE_REQUEST** table. Only particular file requests are updated with **Error** status in the **CI_FILE_REQUEST_DETAIL** table. • If the file upload and process attribute is false, then the requests in the file are uploaded in the application with the **PENDING** status.
Skip Duplicates	• If the skip duplicates attribute is true, then all the duplicate records within a corresponding file will be skipped while uploading in the ORMB staging. • If the skip duplicates attribute is false, then all the available records within a file will be uploaded in ORMB staging.
Service Logs Required	• If the service logs required attribute is true, then service execution details corresponding to the individual request will persist in **CI_FILE_REQUEST_DTL_SERVICE** table. • For every individual request detail in this service table we can capture its identifier details only if we configure a **FK Reference** for the corresponding Service in a **File Request Type**. • In the file request detail dashboard these identifier details are used to navigate to the entity specific user-interface.
File Header Required	• If the file header required attribute is true, then the first row data is passed as a header string to **File Validation Algorithm**. • If the file header required attribute is false, then the first row data will be considered as another request data to be persisted in the ORMB system.
Header XML Tag	• The header xml tag is required, if the **File Format** is **XML** and the **File Transformation Required** attribute is set to true along with the File Header Required attribute. • The header xml tag contains the name of header tag that is used in XML.
File Footer Required	• If the file footer required attribute is true, the last row data is passed as a header string to **File Validation Algorithm**. • If the file footer required attribute is false, the last row data is considered as another request data which is present in the ORMB system.
Footer XML Tag	• The footer xml tag is required, if the **File Format** is **XML** , **File Transform** attribute is set to true and the **File Footer** attribute is set to true. • The footer xml tag contains the name of footer tag that is used in XML.

Copyright © 2024, Oracle and/or its affiliates.

File Request Type (Attribute Name)	Attribute Description
Root XMl Tag	• The root xml tag is required, if the **File Format** is **XML**, and the **File Transform** attribute is set to true. • The root xml tag will have the name of root tag being used in XML.

You can also specify the following soft parameters while executing this batch:

Parameter Name	Description	Mandatory (Yes or No)
File Path	Used to specify the relative path where the file gets uploaded. This path will always begin with either **@SHARED_DIR**or **@INSTALL_DIR** • **@SHARED_DIR** is the configured path of the shared directory. • **@INSTALL_DIR** is the configured path of the installation directory with a property named **spl.runtime.environ.SPLEBASE** within the **spl.properties** file.	Yes
File Name	Used to specify the name of the file (for example CustomerOnboard). If this value is left blank then all the files with the file request type having a corresponding file extension are picked from the given file path. The **File Name** parameter also supports inclusion of the partial value, • If the filename is **CUST%** then the filename that ends with **CUST** must be considered. • If the filename is **%CUST** then the filename that starts with **CUST** must be considered. • If the filename is **%CUST%** then the filename that contains **CUST** must be considered.	Yes
File Request Type	Used to specify the file request type which is uploaded in the system.	Yes
Error Log File Path	Used to specify the CSV file (along with the error logs) that are stored at a provided relative path. This path will always start with either **@SHARED_DIR** or **@INSTALL_DIR**. • **@SHARED_DIR** is the configured path of the shared directory. • **@INSTALL_DIR** is the configured path of the installation directory with a property named **spl.runtime.environ.SPLEBASE** within the **spl.properties** file.	
Execute Batch using File Chunks	Used to specify the execution behavior of the batch. If the parameter value is **Y** then the batch is executed using file chunks.	No
Override Maximum Errors	Used to specify the value to override the maximum number of errors that are allowed before the execution is terminated.	No
Thread Pool Name	Used to specify the thread pool name on which you want to execute the batch.	No

Note: If the **File Transformation and Upload (C1- FTRAN)** batch fails or aborts due to some reason, then you can restart the batch over and over again with the same set of parameters.

Post Execution Check / Clean Up

Once the batch is successfully completed, the logged file details are updated with the **COMPLETE** status. If the **File Upload and Process** attribute value for the corresponding **File Request Type** is set as true, then all of those requests

Copyright © 2024, Oracle and/or its affiliates.

in that file are processed and uploaded with the execution status as either **PROCESSED** or **ERROR** or **SKIPPED** or **INPROGRESS** or **RETRY** or **IGNORE** or **RETRYLIMITEXCEED**

Processing a File Request

The **File Request Processing (C1-FREQP)** batch is used to process the uploaded legacy file records in the staging table. By default the records with **PENDING** and **RETRY** status are processed.

Note: You can execute each **File Request Processing (C1-FREQP)** batch only for a single **File Request Type**.

The **C1-FREQP** batch retrieves all the records with **PENDING** and **RETRY** status, against the **File Request Type** that is provided in the batch. In the case for the records with **RETRY** status, only those records are processed where the **RETRY** count is either less than or equal to the configured **RETRY** count, against its corresponding **REQUEST TYPE**. Before processing any of the file request, the **C1-FREQP** batch validates the following,

- In case the **Error Log File Path** is incorrect
  - The **C1-FREQP** batch is terminated with the **ERROR** status.

Thread iteration strategy is being used for **C1-FREQP** batch. A list of **ThreadWorkUnits** is created on the number of available requests that are present with **PENDING** and **RETRY** status for the priority that is provided in the **CI_FILE_REQUEST_DETAIL** table. Each request is processed for all of the configured services against its corresponding file request type (**<File_Request_Type>**).

Following steps are performed before processing the file request,

If...	**Then...**
If the file request type does not exist	• Error is logged in the **CI_FILE_REQUEST_DTL_MSG** table against the specific request. • Status is updated in the **CI_FILE_REQUEST_DETAIL** table with an **Error** flag.
If the validate request payload flag is true, and the xml payload for every individual service that is configured in the corresponding **<File_Request_Type>** is not provided.	• Status is updated with an **Error** flag. • The error details are logged in **CI_FILE_REQUEST_DTL_MSG** table against the **<Request_ID>** parameter.
If the pre-processing algorithm is configured for the service then invoke the pre-processing algorithm.	• If the **SKIP** flag is set to **TRUE**, then the service is skipped from further execution. • The service status is updated as **SKIPPED** in the **CI_FILE_REQUEST_DTL_SERVICE** table.
If the validation fails while parsing the xml payload	• Error is logged in the **CI_FILE_REQUEST_DTL_MSG** table against the specific request. • The status for the request is updated as **ERROR** in the **CI_FILE_REQUEST_DETAIL** table.

The service used in executing the **C1-FREQP** batch will be executed using the corresponding payload.

If...	**Then...**
If there is a failure while executing the service	• Error is logged in the **CI_FILE_REQUEST_DTL_MSG** table against the specific request. • Status for this request in the **CI_FILE_REQUEST_DETAIL** table is as follows,

Copyright © 2024, Oracle and/or its affiliates.

If...	Then...
	• If there is an error that is configured for **RETRY** status in this specific request for the corresponding file request type, the status is updated as **RETRY**. • If there is an error that is configured for **SENT FOR APPROVAL** status in this specific request for the corresponding file request type, the status is updated as **SFAL**. • If none of the above conditions are satisfied, the status is updated as **Error**. • No other remaining services are executed for this particular file request type. • A rollback operation is performed for every other service that was executed before processing this request.
If the service execution is successful for the payload	• A record is created for the service in the **CI_FILE_REQUEST_DTL_SERVICE** table. • Primary key values will be stamped only if the service has the configuration of the corresponding foreign key (**<FOREIGN_KEY_REF>**) in the request type (**REQUEST_TYPE**), against this service. • The primary key value(s) with its corresponding foreign key is used in the file request dashboard for a 360 degree view. • The status of each configured service is based on the following conditions, • If the status flag of the business object is configured as **TRUE** for the service, then the status of the service is set as **INPROGRESS**. • If the service is skipped, the status is set as **SKIPPED**. • If the service fails with an error, that is configured as **SENT FOR APPROVAL** status in this request for the corresponding file request type, the status of the service is **SFAL**.

Once the request is successfully executed with all of its available services for that specific file request type, the status for that request is updated as **PROCESSED**.

Note:

- The service name is always the root element of its respective **XML** payload.
- The services are executed in a sequence with respect to the given sequence number (**<SEQ_NUM>**) for each service.
- The **<version>** element should not be used in any of the payloads.
- The system would not handle any Java Exceptions or Unknown Errors.

A common API is further exposed to update the service (**<SERVICE>**) status that is either in **INPROGRESS** or **SFAL** status. Once the respective service lifecycle or workflow is completed, this common API updates the status with the available final status. Following are the various types of finally available statuses,

If...	Then...
Execution is successful	Status is set as **PROCESSED**
Execution is canceled	Status is set as **CANCELLED**
Execution is rejected	Status is set as **REJECTED**

Note: Child-parent relationship is also supported in the XML payloads. For two different configured services (**<SERVICE>**) in a single file request type (**<FILE_REQUEST_TYPE>**), an XML payload can be provided with parent-child relationship. This can be achieved by **Dependent Service Name** attribute value.

Copyright © 2024, Oracle and/or its affiliates.

Following is the XML payload for a person-account relationship.

```
<root>
 <request>
  <payload>
   <C1_PERSON_BO>
    <personOrBusiness>PersonName</personOrBusiness>
    <division>CA</division>
    <personName>
     <nameType>PRIM</nameType>
     <entityName>Rudolph1,Jack2</entityName>
     <isPrimaryName>true</isPrimaryName>
    </personName>
    <C1-AccountBO>
     <setupDate>2001-01-01</setupDate>
     <division>CA</divison>
     <accountnumber>
      <isPrimaryId>true</isPrimaryId>
      <accountidentifierType>C1_F_ANO</accountidentifierType>
      <accountNumber>NewAcc2</accountNumber>
     </accountnumber>
     <accountPerson> '
      <accountRelationshipType>MAIN</accountRelationshipType>
      <isFinanciallyResponsible>true</isFinanciallyResponsible>
      <isMainCustomer>true</isMainCustomer>
     </accountPerson>
    </C1-AccountBO>    '
   </C1_PERSON_BO>
  </payload>
 </request>
</root>
```

The use of child-parent relationship in an XML payload, provides the following advantages,

- There is no need to provide any reference of a person (**<PERSON>**) in the account (**<ACCOUNT>**) i.e. for **<accountPerson>** relationship, as the system handles this child-parent condition internally.
- There is no need to have a separate query to fetch the primary key of the parent to map the child in the payload.

Note: The foreign key reference attribute value is required against the **Dependent Schema Name**. Ideally, you must specify the foreign key reference attribute for the **<C1_PERSON_BO>**, as shown in the above example.

There might be some scenarios where the **Dependent Service Name** attribute is not provided, then the XML payload appears as follows,

```
<root>
 <request>
  <payload>
   <C1_PERSON_BO>
    <personOrBusiness>PersonName</personOrBusiness>
    <division>CA</division>
    <personName>
     <nameType>PRIM</nameType>
     <entityName>Rudolph11,Jack22</entityName>
     <isPrimaryName>true</isPrimaryName>
    </personName>
   </C1_PERSON_BO>
  </payload>
  <payload>
   <C1-AccountBO>
     <setupDate>2001-01-01</setupDate>
     <division>CA</divison>
     <accountnumber>
```

Copyright © 2024, Oracle and/or its affiliates.

```
      <isPrimaryId>true</isPrimaryId>
      <accountidentifierType>C1_F_ANO</accountidentifierType>
      <accountNumber>NewAcc2</accountNumber>
     </accountnumber>
     <accountPerson> '
      <accountRelationshipType>MAIN</accountRelationshipType>
      <isFinanciallyResponsible>true</isFinanciallyResponsible>
      <isMainCustomer>true</isMainCustomer>
     </accountPerson>
    </C1-AccountBO>
  </payload>
 </request>
</root>
```

The XML payload mentioned above resembles a multi-threaded batch. It is based on the number of requests that consists of **PENDING** and **RETRY** status available in the staging table, **CI_FILE_REQUEST_DETAIL**, for the supplied **PRIORITY**. You can specify the following parameters while executing the above multi-threaded batch.

Parameter Name	**Description**	**Mandatory (Yes or No)**
File Request Type	Used to specify the records with the **File Request Type** that have the status as either **PENDING** or **RETRY** are fetched for processing.	Yes
Error Log File Path	Used to specify the CSV file (along with the error logs) that are stored at a provided relative path. This path will always start with either **@SHARED_DIR**. • **@SHARED_DIR** is the configured path of the shared directory.	Yes
File Request Status	Used to specify the records for the **File Request Type** that accept input statuses like **PEN** or **RET**.	No
File Name	Used to specify the name of the file (for example CustomerOnboard). If this value is left blank then all the files with the file request type having a corresponding file extension are picked from the given file path. The **File Name** parameter also supports inclusion of the partial value, • If the file-name is**CUST%** then the file-name that ends with **CUST** must be considered. • If the file-name is**%CUST** then the file-name that starts with **CUST** must be considered. • If the file-name is**%CUST%** then the file-name that contains **CUST** must be considered.	No
File Request Upload Date	Used to specify the upload date (in YYYY-MM-DD format) that is considered for processing the file request type.	No
Override Maximum Errors	Used to specify the value to override the maximum number of errors that are allowed before the execution is terminated.	No
Thread Pool Name	Used to specify the thread pool name on which you want to execute the batch.	No

Note: If the **File Request Processing (C1-FREQP)** batch fails or aborts due to some reason, you can restart the batch over and over again with the same set of parameters.

Post Execution Check / Clean Up:

Copyright © 2024, Oracle and/or its affiliates.

Once the batch is successfully completed, the file details are inserted with **COMPLETE** status and the file request details with **PENDING** status.

Note: If there is any change in the **<REQUEST_TYPE>** configuration then it can be replicated either by restarting the thread pool or executing the **Flush All Caches (F1-Flush)** batch.

Archiving a File and/or File Records

Archiving an uploaded file is the process of marking it for long-term storage, ensuring that it remains preserved and accessible for future use or compliance purposes. This helps manage storage efficiently while retaining critical information. The archived files are tagged appropriately within the system, making them easy to identify and retrieve when needed.

Following methods explain the archiving of uploaded file in detail.

- Marking a File for Archival on page 1004
- Marking a File Record for Archival on page 1011

Polling Files on SFTP Server

Polling files on an SFTP server is an automated process in ORMB that ensures the smooth transfer of files from the SFTP server to the ORMB staging area. This process is managed using the **FPOLR** batch (SFTP Poller Batch), the **File Management System**, and specific **Rule Types** that define how files are identified, processed, and monitored.

- **FPOLR Batch (SFTP Poller Batch)** :

  The **FPOLR** batch is responsible for polling files from the SFTP server and determining whether they should be uploaded to ORMB staging. This batch execution is governed by **predefined rules** that filter and categorize files based on business requirements.
- **File Management System** :

  The **File Management System** provides visibility into the status of files uploaded to the SFTP server and ORMB staging. For more information, refer to File Management System on page 952
- **Rule Types for SFTP Poller Batch** :

  The **FPOLR** batch relies on two predefined rule types to control file polling behavior:

  - **File Upload Rule Type (FILE_RQ_TYPE)** :

    This rule type determines which files should be uploaded from the SFTP server to ORMB staging. It ensures only valid files are polled and uploaded. For more information, refer to Rule Types on page 941
  - **File Upload Rule Failure Type (FILE_UP_FAIL)** :

    This rule type handles failures in the file upload process and enables appropriate actions, such as retries, notifications, or manual intervention. For more information, refer to Rule Types on page 941

Prerequisites

To setup the file upload interface, you need to do the following:

- Configure the file upload interface in the master configuration.
- Define a file request type for the file upload interface.
- Set the **C1-FLUPLD** feature configuration.

Related Topics

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
How to edit the master configuration for the file upload interface	Setting the File Upload Interface Master Configuration on page 946
How to define a file request type	Defining a File Request Type on page 966
How to set the **C1-FLUPLD** feature configuration	Setting the C1-FLUPLD Feature Configuration on page 942

Configuring Data Transformation in a File Request Types

Data transformation ensures that file records are transformed as per business requirements before being processed. Configuration involves defining rules, mappings, and attributes for headers, footers, fields, and record validation.

Mapping Header Details for Transformation

Header details in a file can be mapped using the **Data Transformation** section. The header is defined as a block within the file, having a tag name that corresponds to the **Header XML Tag** in the **File Request Type**. For files in **CSV, PSV and TSV** format, the first row of the file represents header of the file. The mapping of header details allows the system to correctly identify and process the file header section.

Mapping Footer Details for Transformation

Similar to the header, footer details are also mapped in the **Data Transformation** section. The last row in a file represent footer in a file. It is represented as a block within the file, with a tag name corresponding to the **Footer XML Tag** in the **File Request Type**. For files in **CSV, PSV and TSV** format, the last row of the file represents footer of the file. Mapping the footer details ensures that validation processes, such as checksum verification, can be performed efficiently.

Mapping Fields for Data Transformation

Fields in a file are mapped based on the **File Request Type - Data Transformation** configuration. The transformation process involves mapping **header**, **footer**, and **field-level data** independently.

There are different approaches to mapping fields depending on the **file format**:

- **CSV, PSV, TSV formats**: Mapping is done using a **sequence index** starting from 1.
- **Fixed Position format**: Mapping is done using **Start Position** and **End Position** values.
- **XML/JSON formats**: Mapping is done using **XPath expressions** specified in the **Source Field Path** field in **Field Transformation**.

Additionally, the **Map Field XPath** feature allows records to be transformed into ORMB-compliant XML structures using a predefined **Data Transformation Algorithm**.

Setting Default Values for a Field

You can configure a default value for a field in the **Data Transformation** section. Default values are primarily used to assign a predefined value when an input field in the file is empty or missing. The default value configuration is done at the **Sequence 0** index in the **Field Transformation** section. For date-type fields, default values can be set using predefined constants like:

Copyright © 2024, Oracle and/or its affiliates.

- `:BUS_DATE` – Business Date
- `:SYSDATE` – System Date
- `:STD_DATE` – Standard Locale Date
- `:BUS_DTTM` – Business Date and Time
- `:SYS_DTTM` – System Date and Time
- `:STD_DTTM` – Standard Locale Date and Time

Setting Mandatory Attribute of a Field

To ensure data integrity, a field can be marked as mandatory within the **Field Transformation** section. If a field is mandatory and does not receive a valid value during file processing, the corresponding record fails validation and is marked as an error. Mandatory attributes are set in the **Data Transformation** settings.

Setting a Date/Time Format for a Field

File upload processing supports multiple date formats, which can be configured in the **Data Transformation** section. The date format is defined at the **File Request Type** level. The system allows configuration of:

- **Client-Specific Date Formats** via a display profile.
- **Custom Date Formats** when a display profile does not support a required format.
- **Multiple Date Formats** within a single **Data Transformation** section.

Setting Record Identifier for Grouping Duplicate Records

To group duplicate records in a file, the **Duplicate Record Identifier Expression** field is used. This expression determines how identical records are identified within a file. When the **Group Duplicate Records** check-box is selected, records that share the same identifier are grouped and processed together. Sorting within groups can be configured using the **Duplicate Records Order By** field, which supports up to two fields in either ascending or descending order.

Skipping BO Level Validations for a Field

BO (Business Object) level validations can be skipped by enabling a **Skip Validation** in **Field Transformation**. This is useful when certain fields do not require ORMB's standard validation checks. Skipping BO-level validations can improve processing performance and prevent unnecessary validation failures for optional fields. For example skipping length check using corresponding field meta-data by file upload batch.

Configuring Static Header and Footer for a File

Static headers and footers can be configured when a file does not contain actual header/footer records but still requires predefined values for validation or processing. This is done under the **Header Transformation** and **Footer Transformation** sections. When enabled:

- The header/footer is used for validation but does not exist in the actual file.
- ORMB uses the configured static values instead of expecting physical header/footer lines in the file.

Copyright © 2024, Oracle and/or its affiliates.

Supported Functions for Default Value

ORMB provides **Default Value Functions** to facilitate data transformation, retrieval, and processing during file uploads. These functions ensure accurate data mapping by fetching values, copying existing data, handling strings, and executing scripts.

List of available functions are:

- FETCH functions
- COPY function
- String Handling
- SCRIPT function
- EVAL function on page 935
- DUPLICATE function on page 936
- Config Function on page 936
- Header Function on page 936
- TO DO Generation on page 937
- Concat Function on page 938

Fetch Functions

These are generic functions that can be used to get the required entity identifiers or its corresponding attributes that persist in ORMB system.

We can get the different field/column values using single fetch function. Provided FETCH functions that has an input parameter **SET_OPERATION** has got the capability to reset the configured corresponding service level operation.

Function inputs can have the LITERAL values i.e. static values enclosed in single quotes or the CONSTANT values i.e. ":BUS_DATE;BUS_DTTM;STD_DATE;STD_DTTM;SYSDATE;SYS_DTTM;CHAR_DATE;FILENAME;FILE_ID" or those configured FIELD names in FIELD TRANSFORMATION whose mapped value will be referred.

Note: To enhance the performance using this fetch function(s) RESULTSET is maintained in that transaction request. So even if the same function is triggered for multiple times with same inputs, we can get the required field value from the RESULTSET in that request transaction.

FETCH functions can be used in only **FIELD TRANSFORMATION** section but not in HEADER and FOOTER TRANSFORMATION.

List of FETCH functions:

- Fetch Account on page 901
- Fetch Account Characteristic on page 902
- Fetch Account Party UID on page 903
- Fetch Adjustment Contract
- Fetch Characteristic Entity Flag on page 906
- Fetch Characteristic Type Flag on page 906
- Fetch Contract on page 906
- Fetch Extendable Lookup value on page 908
- Fetch Invoice Construct ID on page 908
- Fetch Lookup on page 910
- Fetch Parent Person on page 910
- Fetch Price Assignment on page 916
- Fetch Person on page 911
- Fetch Person Characteristic on page 913

Copyright © 2024, Oracle and/or its affiliates.

- Fetch Price Item Contract on page 917
- Fetch Price List on page 918
- Fetch Price List Characteristic on page 920
- Fetch Price List Assignment ID on page 919
- Fetch Person Party UID on page 914
- Fetch Person Relationship on page 915
- Fetch Rate Map Id on page 922
- Fetch User on page 922
- Check Access Group on page 924

Fetch Account

These function can be used to get and map the account identifier or its corresponding column values in ORMB system to the input payload i.e. manipulate the service payload.

Account identifier can be retrieved for either the given **PER_ID_NBR** i.e. with reference to the defined relationship between Person and Account in "Account Person Relationship" table or **ACCT_NBR** i.e. with reference to the relationship defined in the "Maintains External Account Numbers" table.

Other remaining field inputs are optional. Input fields with no mapping can have "NULL" value in the function.

Function:

```
FETCH(ACCT[COLUMN_NAME,PER_ID_NBR,ID_TYPE_CD,ACCT_REL_TYPE_CD,
ACCT_NBR,ACCT_NBR_TYPE_CD,DIVISION,ACCT_USAGE_FLG,PRIM_SW,SET_OPERATION])
```

Expected list of input field/column values:

Input Parameter	Description	Table Description
COLUMN_NAME	Column name to be mapped from the returned RESULTSET. i.e. **OUTPUT** field name	
PER_ID_NBR	Person Id Number	Person Identifier
ID_TYPE_CD	Person Identifier Type	Person Identifier
ACCT_REL_TYPE_CD	Account Relationship Type	Account Person Relationship
ACCT_NBR	Account Identifier	Maintains External Account Numbers
ACCT_NBR_TYPE_CD	Account Identifier Type	Maintains External Account Numbers
DIVISION	Division	Account
ACCT_USAGE_FLG	Account Usage	Account
PRIM_SW	Primary ID switch	Person Identifier
SET_OPERATION	This is a BOOLEAN value. If **TRUE** - Reset operation configured at service level. If **FALSE** - Validate the fetch function output i.e. Throws error if returns NULL or multiple records. If **NULL** – Will do nothing but only throw an error if function return multiple records.	

Copyright © 2024, Oracle and/or its affiliates.

List of output field/column values:

Output Parameter	Description	Table Description
ACCT_ID	Account ID	Account
CIS_DIVISION	Division of an Account	Account
CURRENCY_CD	Currency Code of an Account	Account
PER_ID	Person ID (Returned only when PER_ID_NBR is passed as an function input)	Person Identifier
ID_TYPE_CD	Person Identifier Type (Returned only when PER_ID_NBR is passed as an function input)	Person Identifier
ACCT_REL_TYPE_CD	Account Relationship Type (Returned only when PER_ID_NBR is passed as an function input)	Account Person Relationship
ACCT_NBR_TYPE_CD	Account Identifier Type	Maintains External Account Numbers
ACCT_USAGE_FLG	Account Usage	Account
VERSION	Version	Account

Fetch Account Characteristic

These function can be used to get and map the account Identifier or its corresponding Characteristic values in ORMB system to the input payload i.e. manipulate the service payload.

ACCT_NBR input is mandatory and the other remaining field inputs are optional. Input fields with no mapping can have “NULL” value in the function.

Function:

```
FETCH(ACCT_CHAR[COLUMN_NAME,ACCT_NBR,ACCT_NBR_TYPE_CD,CHAR_TYPE_CD,
DIVISION,ACCT_USAGE_FLG,PRIM_SW,SET_OPERATION])
```

Expected list of input field/column values:

Input Parameter	Description	Table Description
COLUMN_NAME	Column name to be mapped from the returned RESULTSET. i.e. **OUTPUT** field name	
ACCT_NBR	Account Identifier	Maintains External Account Numbers
ACCT_NBR_TYPE_CD	Account Identifier Type	Maintains External Account Numbers
CHAR_TYPE_CD	Characteristic Type Code	Account Characteristics
DIVISION	Division	Account
ACCT_USAGE_FLG	Account Usage	Account
PRIM_SW	Primary ID switch	Person Identifier
SET_OPERATION	This is a BOOLEAN value.	

Copyright © 2024, Oracle and/or its affiliates.

	If **TRUE** - Reset operation configured at service level. If **FALSE** - Validate the fetch function output i.e. Throws error if returns NULL or multiple records. If **NULL** – Will do nothing but only throw an error if function return multiple records.	

List of output field/column values:

Output Parameter	Description	Table Description
ACCT_ID	Account ID	Account
CIS_DIVISION	Division of an Account	Account
CURRENCY_CD	Currency Code of an Account	Account
ACCT_NBR_TYPE_CD	Account Identifier Type	Maintains External Account Numbers
ACCT_USAGE_FLG	Account Usage	Account
CHAR_TYPE_CD	Characteristic Type Code	Account Characteristics
CHAR_VAL	Characteristic Value	Account Characteristics
EFFDT	Effective Date	Account Characteristics
CHAR_VAL_FK1	Characteristic Value FK1	Account Characteristics
CHAR_VAL_FK2	Characteristic Value FK2	Account Characteristics
CHAR_VAL_FK3	Characteristic Value FK3	Account Characteristics
CHAR_VAL_FK4	Characteristic Value FK4	Account Characteristics
CHAR_VAL_FK5	Characteristic Value FK5	Account Characteristics
VERSION	Version	Account

Fetch Account Party UID

These function can be used to get and map the account identifier, Party UID or its corresponding column values in ORMB system to the input payload i.e. manipulate the service payload.

Account identifier can be retrieved for either the given **PER_ID_NBR** i.e. with reference to the defined relationship between Person and Account in “Account Person Relationship” table or **ACCT_NBR** i.e. with reference to the relationship defined in the “Maintains External Account Numbers” table.

Other remaining field inputs are optional. Input fields with no mapping can have “NULL” value in the function.

Function:

```
FETCH(ACCT_PUID[COLUMN_NAME,PER_ID_NBR,ID_TYPE_CD,ACCT_REL_TYPE_CD,
ACCT_NBR,ACCT_NBR_TYPE_CD,ACCT_ID,DIVISION,ACCT_USAGE_FLG,PRIM_SW,SET_OPERATION])
```

Expected list of input field/column values:

Input Parameter	Description	Table Description

Copyright © 2024, Oracle and/or its affiliates.

COLUMN_NAME	Column name to be mapped from the returned RESULTSET. i.e. **OUTPUT** field name	
PER_ID_NBR	Person Id Number	Person Identifier
ID_TYPE_CD	Person Identifier Type	Person Identifier
ACCT_REL_TYPE_CD	Account Relationship Type	Account Person Relationship
ACCT_NBR	Account Identifier	Maintains External Account Numbers
ACCT_NBR_TYPE_CD	Account Identifier Type	Maintains External Account Numbers
DIVISION	Division	Account
ACCT_USAGE_FLG	Account Usage	Account
PRIM_SW	Primary ID switch	Person Identifier
SET_OPERATION	This is a BOOLEAN value. If **TRUE** - Reset operation configured at service level. If **FALSE** - Validate the fetch function output i.e. Throws error if returns NULL or multiple records. If **NULL** – Will do nothing but only throw an error if function return multiple records.	

List of output field/column values:

Output Parameter	**Description**	**Table Description**
ACCT_ID	Account ID	Account
CIS_DIVISION	Division of an Account	Account
CURRENCY_CD	Currency Code of an Account	Account
PER_ID	Person ID	Person Identifier
PARTY_UID	Party UID	Party
ACCT_REL_TYPE_CD	Account Relationship Type	Account Person Relationship
ACCT_NBR_TYPE_CD	Account Identifier Type	Maintains External Account Numbers
ACCT_USAGE_FLG	Account Usage	Account
VERSION	Version	Account

Fetch Adjustment Contract

These function can be used to get and map the account identifier, Contract ID or its corresponding column values in ORMB system to the input payload i.e. manipulate the service payload.

Get the Contract ID, Contract Type for the relationship between “SA Type Adjustment Type Profile” and “Adjustment Type / Adjustment Type Profile”

Copyright © 2024, Oracle and/or its affiliates.

ACCT_NBR is mandatory input and other remaining field inputs are optional. Input fields with no mapping can have "NULL" value in the function.

Function:

```
FETCH(ADJSA[COLUMN_NAME,ACCT_NBR,ACCT_NBR_TYPE_CD,ADJ_TYP_CD,
DIVISION,ACCT_USAGE_FLG,SET_OPERATION])
```

Expected list of input field/column values:

Input Parameter	Description	Table Description
COLUMN_NAME	Column name to be mapped from the returned RESULTSET. i.e. **OUTPUT** field name	
ACCT_NBR	Account Identifier	Maintains External Account Numbers
ACCT_NBR_TYPE_CD	Account Identifier Type	Maintains External Account Numbers
ADJ_TYP_CD	Adjustment Type Code	Adjustment Type / Adjustment Type Profile
DIVISION	Division	Account
ACCT_USAGE_FLG	Account Usage	Account
SET_OPERATION	This is a BOOLEAN value. If **TRUE** - Reset operation configured at service level. If **FALSE** - Validate the fetch function output i.e. Throws error if returns NULL or multiple records. If **NULL** – Will do nothing but only throw an error if function return multiple records.	

List of output field/column values:

Output Parameter	Description	Table Description
ACCT_ID	Account ID	Account
CIS_DIVISION	Division of an Account	Account
CURRENCY_CD	Currency Code of an Account	Account
SA_ID	Contract ID	SA (Service Agreement)
SA_TYPE_CD	Contract Type	SA (Service Agreement)
ADJ_TYPE_CD	Account Relationship Type	Adjustment Type / Adjustment Type Profile
ACCT_NBR_TYPE_CD	Account Identifier Type	Maintains External Account Numbers
ACCT_USAGE_FLG	Account Usage	Account

Copyright © 2024, Oracle and/or its affiliates.

VERSION	Version	SA (Service Agreement)

Fetch Characteristic Entity Flag

These function can be used to get the Characteristic Entity Flag for the given Characteristic Type Code.

CHAR_TYPE is mandatory input for this function.

Function:

```
FETCH(CHAR_ENTITY_FLG[CHAR_TYPE])
```

Expected list of input field/column values:

Input Parameter	Description	Table Description
CHAR_TYPE_CD	Characteristic Type Code	Characteristic Type Entity

List of output field/column values:

Output Parameter	Description	Table Description
CHAR_ENTITY_FLG	Characteristic Entity	Characteristic Type Entity

Fetch Characteristic Type Flag

These function can be used to get the Type of CHAR value for the given Characteristic Type Code.

CHAR_TYPE is mandatory input for this function.

Function:

```
FETCH(CHAR_TYPE_FLG[CHAR_TYPE])
```

Expected list of input field/column values:

Input Parameter	Description	Table Description
CHAR_TYPE_CD	Characteristic Type Code	Characteristic Type

List of output field/column values:

Output Parameter	Description	Table Description
CHAR_TYPE_FLG	Type of Char Value	Characteristic Type

Fetch Contract

These function can be used to get the Contract ID, Contract Type and other corresponding below listed column values.

Contract identifier can be retrieved using the Contract and Account relationship. Account identifier can be retrieved for either the given **PER_ID_NBR** i.e. with reference to the defined relationship between Person and Account in “Account Person Relationship” table or **ACCT_NBR** i.e. with reference to the relationship defined in the “Maintains External Account Numbers” table.

Either of the inputs **PER_ID_NBR or ACCT_NBR** is mandatory input for this function.

Function:

```
FETCH(SA[COLUMN_NAME,PER_ID_NBR,ID_TYPE_CD,ACCT_REL_TYPE_CD,ACCT_NBR,
ACCT_NBR_TYPE_CD,SA_TYPE_CD,DIVISION,ACCT_USAGE_FLG,PRIM_SW,SET_OPERATION])
```

Copyright © 2024, Oracle and/or its affiliates.

Expected list of input field/column values:

Input Parameter	Description	Table Description
COLUMN_NAME	Column name to be mapped from the returned RESULTSET. i.e. **OUTPUT** field name	
PER_ID_NBR	Person Id Number	Person Identifier
ID_TYPE_CD	Person Identifier Type	Person Identifier
ACCT_REL_TYPE_CD	Account Relationship Type	Account Person Relationship
ACCT_NBR	Account Identifier	Maintains External Account Numbers
ACCT_NBR_TYPE_CD	Account Identifier Type	Maintains External Account Numbers
SA_TYPE_CD	Contract Type	SA (Service Agreement)
DIVISION	Division	Account
ACCT_USAGE_FLG	Account Usage	Account
PRIM_SW	Primary ID switch	Person Identifier
SET_OPERATION	This is a BOOLEAN value. If **TRUE** - Reset operation configured at service level. If **FALSE** - Validate the fetch function output i.e. Throws error if returns NULL or multiple records. If **NULL** – Will do nothing but only throw an error if function return multiple records.	

List of output field/column values:

Output Parameter	Description	Table Description
ACCT_ID	Account ID	Account
CIS_DIVISION	Division of an Account	Account
CURRENCY_CD	Currency Code of an Account	Account
PER_ID	Person ID (Returned only when PER_ID_NBR is passed as an function input)	Person Identifier
SA_ID	Contract ID	SA (Service Agreement)
ACCT_REL_TYPE_CD	Account Relationship Type (Returned only when PER_ID_NBR is passed as an function input)	Account Person Relationship
ACCT_NBR_TYPE_CD	Account Identifier Type	Maintains External Account Numbers
SA_TYPE_CD	Contract Type	SA (Service Agreement)

Copyright © 2024, Oracle and/or its affiliates.

ACCT_USAGE_FLG	Account Usage	Account
VERSION	Version	Account

Fetch Extendable Lookup value

These function can be used to get the value from the extendable lookup's CLOB field i.e. XML payload. Last parameter in function i.e. **VALIDATE** is a Boolean parameter. If **TRUE**, it will throw an error, if no value is returned for this function. **FALSE** will do nothing.

It has two filter criteria to get the required field value.

FILTER 1 – To get the extendable lookup CLOB object i.e. XML payload

Parameters **BUS_OBJ_CD, EXT_LOOKUP_VALUE** and **EXT_LOOKUP_USAGE_FLG** can be used to get the required extendable lookup record.

Parameters **BUS_OBJ_CD** and **EXT_LOOKUP_VALUE** are mandatory inputs and **EXT_LOOKUP_USAGE_FLG** is optional for this function.

FILTER 2 – To get the required element value from this retrieved CLOB object i.e. XML payload

You can provide filter criteria to get the specific element value in the XML payload. Criteria can be defined within **INPUT{}** block. Comma-Separated multiple inputs can be provided.

These can have criteria in following format,

ELEMENT_REF_XPATH=INPUT_VALUE

ELEMENT_REF_XPATH – This will be the element XPATH in the retrieved XML payload of extendable lookup for which you want to filter the XML block data.

INPUT_VALUE - This will be the element on which you want to filter the XML block data. This can be a LITERAL value or you can map the configured FIELD name in FIELD TRANSFORMATION.

Function:

```
FETCH(EXT_LOOKUP[BUS_OBJ_CD,EXT_LOOKUP_VALUE,EXT_LOOKUP_USAGE_FLG,
INPUT{LOOKUP_XPATH1=?;LOOKUP_XPATH2=?},OUTPUT_XPATH,VALIDATE])
```

Expected list of input field/column values:

Input Parameter	Description	Table Description
BUS_OBJ_CD	Business Object Code	
EXT_LOOKUP_VALUE	Extendable Lookup Value	
EXT_LOOKUP_USAGE_FLG	Extendable Lookup Usage flag	

List of output field/column values:

OUTPUT_XPATH – This parameter will be the XPATH of the ELEMENT in XML payload of extendable lookup CLOB object. Value of this element will be returned as a result.

Fetch Invoice Construct ID

These function can be used to get the Invoice Construct Identifier and other corresponding below listed column/output values.

Invoice Contract identifier can be retrieved using Account Identifier i.e. ACCT_NBR. Using this identifier we look for BILLING ACCOUNT ID in Construct table.

ACCT_NBR is mandatory input while the other remaining input parameters are optional for this function.

Copyright © 2024, Oracle and/or its affiliates.

Function:

```
FETCH(INV_CONSTRUCT[COLUMN_NAME,ACCT_NBR,ACCT_NBR_TYPE_CD,ACCT_ID,
TEMPLATE_PURPOSE_FLG,EFFT_START_DT,EFFT_END_DT,SET_OPERATION])
```

Expected list of input field/column values:

Input Parameter	**Description**	**Table Description**
COLUMN_NAME	Column name to be mapped from the returned RESULTSET. i.e. **OUTPUT** field name	
ACCT_NBR	Account Identifier	Maintains External Account Numbers
ACCT_NBR_TYPE_CD	Account Identifier Type	Maintains External Account Numbers
ACCT_ID	Account ID	Maintains External Account Numbers
TEMPLATE_PURPOSE_FLG	Template Purpose Flag	Construct Template
EFFT_START_DT	Effective Start Date	Construct
EFFT_END_DT	Effective End Date	Construct
STATUS_FLG	Status Flag. By default its value will be 'ACTV'.	Construct
SET_OPERATION	This is a BOOLEAN value. If **TRUE** - Reset operation configured at service level. If **FALSE** - Validate the fetch function output i.e. Throws error if returns NULL or multiple records. If **NULL** – Will do nothing but only throw an error if function return multiple records.	

List of output field/column values:

Output Parameter	**Description**	**Table Description**
CONSTRUCT_ID	Construct ID	Construct
TEMPLATE_PURPOSE_FLG	Template Purpose Flag	Construct Template
EFFT_START_DT	Effective Start Date	Construct
EFFT_END_DT	Effective End Date	Construct
CONSTRUCT_DESC	Contract Description	Construct
TYPE_FLG	Type Flag	Construct
VERSION_NUM	Version Number	Construct
BILLING_ACCT_ID	Contract Type	Construct Template

Copyright © 2024, Oracle and/or its affiliates.

STATUS_FLG	Status Flag	Construct

Fetch Lookup

These function can be used to get the lookup values.

FIELD_NAME is mandatory input while the other remaining input parameters are optional for this function.

Function:

```
FETCH(LOOKUP[COLUMN_NAME,FIELD_NAME,FIELD_VALUE,DESCR,LANGUAGE_CD])
```

Expected list of input field/column values:

Input Parameter	Description	Table Description
COLUMN_NAME	Column name to be mapped from the returned RESULTSET. i.e. **OUTPUT** field name	
FIELD_NAME	Lookup Field Name	Lookup Field Value Language
FIELD_VALUE	Lookup Field Value	Lookup Field Value Language
DESCR	Description	Lookup Field Value Language
LANGUAGE_CD	Language Code	Lookup Field Value Language

List of output field/column values:

Output Parameter	Description	Table Description
FIELD_VALUE	Lookup Field Value	Lookup Field Value Language
DESCR	Description	Lookup Field Value Language
DESCRLONG	Detail Description	Lookup Field Value Language

Fetch Parent Person

These function can be used to get the root parent person from the person hierarchy

Account identifier can be retrieved for either the given **PER_ID_NBR** i.e. with reference to the defined relationship between Person and Account in “Account Person Relationship” table or **ACCT_NBR** i.e. with reference to the relationship defined in the “Maintains External Account Numbers” table.

Only input parameters **PER_ID_NBR** and **ID_TYPE_CD** are optional and remaining are mandatory for this function.

Function:

```
FETCH(PARENT_PER[COLUMN_NAME,PER_ID_NBR,ID_TYPE_CD,PER_REL_TYPE_CD,
ACCT_REL_TYPE_CD,ACCT_NBR,ACCT_NBR_TYPE_CD,SET_OPERATION])
```

Expected list of input field/column values:

Input Parameter	Description	Table Description
COLUMN_NAME	Column name to be mapped from the returned RESULTSET. i.e. **OUTPUT** field name	
PER_ID_NBR	Person Id Number	Person Identifier
ID_TYPE_CD	Person Identifier Type	Person Identifier

Copyright © 2024, Oracle and/or its affiliates.

PER_REL_TYPE_CD	Person Relationship Type Code	Person Identifier
ACCT_REL_TYPE_CD	Account Relationship Type	Account Person Relationship
ACCT_NBR	Account Identifier	Maintains External Account Numbers
ACCT_NBR_TYPE_CD	Account Identifier Type	Maintains External Account Numbers
SET_OPERATION	This is a BOOLEAN value. If **TRUE** - Reset operation configured at service level. If **FALSE** - Validate the fetch function output i.e. Throws error if returns NULL or multiple records. If **NULL** – Will do nothing but only throw an error if function return multiple records.	

List of output field/column values:

Output Parameter	Description	Table Description
PER_ID	Person ID (Returned only when PER_ID_NBR is passed as an function input)	Person Identifier
PER_ID_NBR	Person Id Number	Person Identifier
ID_TYPE_CD	Person Identifier Type	Person Identifier
PRIM_SW	Primary ID switch	Person Identifier
VERSION	Version	Person Identifier

Fetch Person

These function can be used to get the Person identifier with its corresponding detail field values i.e. below listed column/output values.

Account identifier can be retrieved for either the given **PER_ID_NBR** i.e. with reference to the defined relationship between Person and Account in “Account Person Relationship” table or **ACCT_NBR** i.e. with reference to the relationship defined in the “Maintains External Account Numbers” table.

Either of the inputs **PER_ID_NBR or ACCT_NBR** is mandatory input for this function.

Function:

```
FETCH(PER[COLUMN_NAME,PER_ID_NBR,ID_TYPE_CD,ACCT_REL_TYPE_CD,ACCT_NBR,
ACCT_NBR_TYPE_CD,DIVISION,ACCT_USAGE_FLG,PRIM_SW,SET_OPERATION])
```

Expected list of input field/column values:

Input Parameter	Description	Table Description
COLUMN_NAME	Column name to be mapped from the returned RESULTSET. i.e. **OUTPUT** field name	

Copyright © 2024, Oracle and/or its affiliates.

PER_ID_NBR	Person Id Number	Person Identifier
ID_TYPE_CD	Person Identifier Type	Person Identifier
ACCT_REL_TYPE_CD	Account Relationship Type	Account Person Relationship
ACCT_NBR	Account Identifier	Maintains External Account Numbers
ACCT_NBR_TYPE_CD	Account Identifier Type	Maintains External Account Numbers
DIVISION	Division	Account
ACCT_USAGE_FLG	Account Usage	Account
PRIM_SW	Primary ID switch	Person Identifier
SET_OPERATION	This is a BOOLEAN value. If **TRUE** - Reset operation configured at service level. If **FALSE** - Validate the fetch function output i.e. Throws error if returns NULL or multiple records. If **NULL** – Will do nothing but only throw an error if function return multiple records.	

List of output field/column values:

Output Parameter	**Description**	**Table Description**
PER_ID	Person ID	Person Identifier
ID_TYPE_CD	Person Identifier Type (Returned only when PER_ID_NBR is passed as an function input)	Person Identifier
ACCESS_GRP_CD	Access Group Code (Returned only when PER_ID_NBR is passed as an function input)	Person
PRIM_SW	Primary ID (Returned only when PER_ID_NBR is passed as an function input)	Person Identifier
ACCT_ID	Account ID (Returned only when ACCT_NBR is passed as an function input)	Account
CIS_DIVISION	Division of an Account (Returned only when ACCT_NBR is passed as an function input)	Account
CURRENCY_CD	Currency Code of an Account (Returned only when ACCT_NBR is passed as an function input)	Account

Copyright © 2024, Oracle and/or its affiliates.

ACCT_REL_TYPE_CD	Account Relationship Type (Returned only when ACCT_NBR is passed as an function input)	Account Person Relationship
ACCT_NBR_TYPE_CD	Account Identifier Type (Returned only when ACCT_NBR is passed as an function input)	Maintains External Account Numbers
ENTITY_NAME	Contract Type (Returned only when ACCT_NBR is passed as an function input)	Person Name
ACCT_USAGE_FLG	Account Usage (Returned only when ACCT_NBR is passed as an function input)	Account
VERSION	Version	Person

Fetch Person Characteristic

These function can be used to get the Person identifier and Characteristics along with its other corresponding field values i.e. below listed column/output values.

Parameters **PER_ID_NBR and CHAR_TYPE_CD** are mandatory inputs for this function.

Function:

```
FETCH(PER_CHAR[COLUMN_NAME,PER_ID_NBR,ID_TYPE_CD,CHAR_TYPE_CD,PRIM_SW,
SET_OPERATION])
```

Expected list of input field/column values:

Input Parameter	Description	Table Description
COLUMN_NAME	Column name to be mapped from the returned RESULTSET. i.e. **OUTPUT** field name	
PER_ID_NBR	Person Id Number	Person Identifier
ID_TYPE_CD	Person Identifier Type	Person Identifier
CHAR_TYPE_CD	Characteristic Type Code	Account Person Relationship
PRIM_SW	Primary ID switch	Person Identifier
SET_OPERATION	This is a BOOLEAN value. If **TRUE** - Reset operation configured at service level. If **FALSE** - Validate the fetch function output i.e. Throws error if returns NULL or multiple records. If **NULL** – Will do nothing but only throw an error if function return multiple records.	

List of output field/column values:

Output Parameter	Description	Table Description

Copyright © 2024, Oracle and/or its affiliates.

PER_ID	Person ID	Person Identifier
ID_TYPE_CD	Person Identifier Type	Person Identifier
ACCESS_GRP_CD	Access Group Code	Person
PRIM_SW	Primary ID	Person Identifier
CHAR_TYPE_CD	Characteristic Type Code	Person Characteristics
CHAR_VAL	Characteristic Value	Person Characteristics
EFFDT	Effective Date	Person Characteristics
CHAR_VAL_FK1	Characteristic Value FK1	Person Characteristics
CHAR_VAL_FK2	Characteristic Value FK2	Person Characteristics
CHAR_VAL_FK3	Characteristic Value FK3	Person Characteristics
CHAR_VAL_FK4	Characteristic Value FK4	Person Characteristics
CHAR_VAL_FK5	Characteristic Value FK5	Person Characteristics
VERSION	Version	Person

Fetch Person Party UID

These function can be used to get and map the Person identifier, Party UID or its corresponding column values in ORMB system to the input payload i.e. manipulate the service payload.

Parameter **PER_ID_NBR** is mandatory input for this function.

Function:

```
FETCH(ACCT_PUID[COLUMN_NAME,PER_ID_NBR,ID_TYPE_CD,DIVISION,ACCT_USAGE_FLG,
PRIM_SW,SET_OPERATION])
```

Expected list of input field/column values:

Input Parameter	**Description**	**Table Description**
COLUMN_NAME	Column name to be mapped from the returned RESULTSET. i.e. **OUTPUT** field name	
PER_ID_NBR	Person Id Number	Person Identifier
ID_TYPE_CD	Person Identifier Type	Person Identifier
DIVISION	Division	Account
ACCT_USAGE_FLG	Account Usage	Account
PRIM_SW	Primary ID switch	Person Identifier
SET_OPERATION	This is a BOOLEAN value. If **TRUE** - Reset operation configured at service level. If **FALSE** - Validate the fetch function output i.e. Throws error if returns NULL or multiple records.	

Copyright © 2024, Oracle and/or its affiliates.

	If **NULL** – Will do nothing but only throw an error if function return multiple records.	

List of output field/column values:

Output Parameter	Description	Table Description
CIS_DIVISION	Division of an Account	Account
CURRENCY_CD	Currency Code of an Account	Account
PER_ID	Person ID	Person Identifier
PARTY_UID	Party UID	Party
ACCT_REL_TYPE_CD	Account Relationship Type	Account Person Relationship
ACCT_USAGE_FLG	Account Usage	Account
VERSION	Version	Party

Fetch Person Relationship

These function can be used to get and map the Person to Person relationship with its corresponding column/output values listed below.

Parameter **PER_ID_NBR** is mandatory input for this function.

Function:

```
FETCH(PER_RELATION[COLUMN_NAME,PER_ID1,PER_ID2,PER_REL_TYPE_CD,FINAN_REL_SW,
END_DT,SET_OPERATION])
```

Expected list of input field/column values:

Input Parameter	Description	Table Description
COLUMN_NAME	Column name to be mapped from the returned RESULTSET. i.e. **OUTPUT** field name	
PER_ID1	Person1	Person to Person
PER_ID2	Child Person	Person to Person
PER_REL_TYPE_CD	Relationship Type	Person to Person
FINAN_REL_SW	Financial Relationship	Person to Person
END_DT	End Date (This date will be compared to be greater than or equal to END DATE or retrieved if NULL)	Person to Person
SET_OPERATION	This is a BOOLEAN value. If **TRUE** - Reset operation configured at service level. If **FALSE** - Validate the fetch function output i.e. Throws error if returns NULL or multiple records.	

Copyright © 2024, Oracle and/or its affiliates.

	If **NULL** – Will do nothing but only throw an error if function return multiple records.	

List of output field/column values:

Output Parameter	Description	Table Description
PER_ID1	Person1	Person to Person
PER_ID2	Child Person	Person to Person
PER_REL_TYPE_CD	Relationship Type	Person to Person
FINAN_REL_SW	Financial Relationship	Person to Person
END_DT	End Date	Person to Person
VERSION	Version	Person to Person

Fetch Price Assignment

These function can be used to get the price assignment details for the given Owner ID or Price Item Code or Price Assign ID.

Function:

```
FETCH(PRICE_ASSIGNMENT[COLUMN_NAME,OWNER_ID,PRICEITEM_CD,PRICE_STATUS_FLAG,
CRR_DT,PRICE_ASGN_ID,SET_OPERATION])
```

Expected list of input field/column values:

Input Parameter	Description	Table Description
COLUMN_NAME	Column name to be mapped from the returned RESULTSET. i.e. **OUTPUT** field name	
OWNER_ID	Owner ID	Price Assignment
PRICEITEM_CD	Price Item	Price Assignment
PRICE_STATUS_FLAG	Pricing Status	Price Assignment
CRR_DT	This date will be compared to be greater than or equal to END DATE in Price Assignment	Price Assignment
PRICE_ASGN_ID	Price Assignment ID	Price Assignment
SET_OPERATION	This is a BOOLEAN value. If **TRUE** - Reset operation configured at service level. If **FALSE** - Validate the fetch function output i.e. Throws error if returns NULL or multiple records. If **NULL** – Will do nothing but only throw an error if function return multiple records.	

Copyright © 2024, Oracle and/or its affiliates.

List of output field/column values:

Output Parameter	Description	Table Description
PRICE_ASGN_ID	Price Assignment ID	Price Assignment
OWNER_ID	Owner ID	Price Assignment
PA_OWNER_TYPE_FLG	Price Assignment Owner Type Flag	Price Assignment
START_DT	Start Date	Price Assignment
END_DT	End Date	Price Assignment
PRICEITEM_CD	Price Item	Price Assignment
PRICE_CURRENCY_CD	Pricing Currency	Price Assignment
PRICE_STATUS_FLAG	Pricing Status	Price Assignment
RS_CD	Rate Schedule	Price Assignment
PA_TYPE_FLAG	Price Assignment Type	Price Assignment
TXN_RATING_CRITERIA	Transaction Rating Criteria	Price Assignment

Fetch Price Item Contract

These function can be used to get the price item for contract along with its other corresponding field values i.e. below listed column/output values.

Parameters **ACCT_NBR** is mandatory input for this function.

Function:

```
FETCH(PISA[COLUMN_NAME,ACCT_NBR,ACCT_NBR_TYPE_CD,PRICE_ITEM_CD,DIVISION,
ACCT_USAGE_FLG,SET_OPERATION])
```

Expected list of input field/column values:

Input Parameter	Description	Table Description
COLUMN_NAME	Column name to be mapped from the returned RESULTSET. i.e. **OUTPUT** field name	
ACCT_NBR	Account Identifier	Maintains External Account Numbers
ACCT_NBR_TYPE_CD	Account Identifier Type	Maintains External Account Numbers
DIVISION	Division	Account
ACCT_USAGE_FLG	Account Usage	Account
PRIM_SW	Primary ID switch	Person Identifier
PRICE_ITEM_CD	Price Item	Pricing Item
SET_OPERATION	This is a BOOLEAN value. If **TRUE** - Reset operation configured at service level.	

Copyright © 2024, Oracle and/or its affiliates.

	If **FALSE** - Validate the fetch function output i.e. Throws error if returns NULL or multiple records. If **NULL** – Will do nothing but only throw an error if function return multiple records.	

List of output field/column values:

Output Parameter	Description	Table Description
ACCT_ID	Account ID	Account
CIS_DIVISION	Division of an Account	Account
CURRENCY_CD	Currency Code of an Account	Account
ACCT_NBR_TYPE_CD	Account Identifier Type	Maintains External Account Numbers
ACCT_USAGE_FLG	Account Usage	Account
SA_ID	Contract ID	SA (Service Agreement)
SA_TYPE_CD	Contract Type	SA (Service Agreement)
VERSION	Version	SA (Service Agreement)

Fetch Price List

These function can be used to get the price list along with its other corresponding field values i.e. below listed column/output values.

Either of parameter **PRICELIST_ID or DESCR** is mandatory inputs for this function.

Function:

```
FETCH(PRICE_LIST[COLUMN_NAME,PRICELIST_ID,PL_TYPE,PL_STATUS_FLG,DESCR,
LANGUAGE_CD,SET_OPERATION])
```

Expected list of input field/column values:

Input Parameter	Description	Table Description
COLUMN_NAME	Column name to be mapped from the returned RESULTSET. i.e. **OUTPUT** field name	
PRICELIST_ID	Price List ID	Price List
PL_TYPE	Price List Type	Price List
PL_STATUS_FLG	Status	Price List
DESCR	Description (This can have the wild card ‘%’ to pass the partial description)	Price List
LANGUAGE_CD	Primary ID switch	Price List Language
SET_OPERATION	This is a BOOLEAN value.	

Copyright © 2024, Oracle and/or its affiliates.

	If **TRUE** - Reset operation configured at service level. If **FALSE** - Validate the fetch function output i.e. Throws error if returns NULL or multiple records. If **NULL** – Will do nothing but only throw an error if function return multiple records.	

List of output field/column values:

Output Parameter	Description	Table Description
PRICELIST_ID	Price List ID	Price List
PL_TYPE	Price List Type	Price List
PL_STATUS_FLG	Status	Price List
DESCR	Description	Price List Language
PL_START_DT	Effective Start Date	Price List
PL_END_DT	Effective End Date	Price List
PL_GLOBAL_SW	Global Price List Switch	Price List

Fetch Price List Assignment ID

These function can be used to get the active price list assignment identifier along with its other corresponding field values i.e. below listed column/output values.

Different combination of inputs can be used to invoke this function.

Either parameters **ACCT_NBR** and **ACCT_NBR_TYPE_CD** can be passed to get the **PARTY_UID** or directly pass the **PARTY_UID** input to get the required active price list assignment details.

Either parameters **DESCR** and **LANGUAGE_CD** can be passed to get the **PRICELIST_ID** or directly pass the **PRICELIST_ID** input to get the required active price list assignment details.

Function:

```
FETCH(PL_ASSGN[COLUMN_NAME,ACCT_NBR,ACCT_NBR_TYPE_CD,PARTY_UID,PRICELIST_ID,
DESCR,LANGUAGE_CD,START_DT,END_DT, PL_STATUS_FLG,SET_OPERATION])
```

Expected list of input field/column values:

Input Parameter	Description	Table Description
COLUMN_NAME	Column name to be mapped from the returned RESULTSET. i.e. **OUTPUT** field name	
ACCT_NBR	Account Identifier	Maintains External Account Numbers
ACCT_NBR_TYPE_CD	Account Identifier Type	Maintains External Account Numbers
PARTY_UID	Party UID	Price List assignment
PRICELIST_ID	Price List ID	Price List

Copyright © 2024, Oracle and/or its affiliates.

DESCR	Price List Description (This can have the wild card '%' to pass the partial description)	Price List Language
LANGUAGE_CD	Language Code (Default is set to 'ENG')	Price List Language
START_DT	Start Date (This date will be compared to be less than or equal to START DATE in Price List assignment)	Price List assignment
END_DT	End Date (This date will be compared to be greater than or equal to END DATE in Price List assignment)	Price List assignment
PL_STATUS_FLG	Status Flag. By default its value will be 'ACTV'.	Price List
SET_OPERATION	This is a BOOLEAN value. If **TRUE** - Reset operation configured at service level. If **FALSE** - Validate the fetch function output i.e. Throws error if returns NULL or multiple records. If **NULL** – Will do nothing but only throw an error if function return multiple records.	

List of output field/column values:

Output Parameter	**Description**	**Table Description**
PRICELIST_ASGN_ID	Account ID	Price List assignment
PARTY_UID	Party UID	Price List assignment
PRICELIST_ID	Price List ID	Price List assignment
START_DT	Start Date	Price List assignment
END_DT	End Date	Price List assignment
BUS_OBJ_CD	Business Object Code	Price List assignment
SEQNO	Sequence No.	Price List assignment

Fetch Price List Characteristic

These function can be used to get the Price List identifier and its Characteristics along with its other corresponding field values i.e. below listed column/output values.

Either of parameters **CHAR_VAL** or **CHAR_TYPE_CD** or **ADHOC_CHAR_VAL** is mandatory input for this function.

Function:

```
FETCH(PRICE_LIST_CHAR[COLUMN_NAME,CHAR_TYPE_CD,ADHOC_CHAR_VAL,CHAR_VAL,
PRICELIST_ID,PL_TYPE,PL_STATUS_FLG,SET_OPERATION])
```

Copyright © 2024, Oracle and/or its affiliates.

Expected list of input field/column values:

Input Parameter	Description	Table Description
COLUMN_NAME	Column name to be mapped from the returned RESULTSET. i.e. **OUTPUT** field name	
CHAR_TYPE_CD	Characteristic Type Code	Price List Characteristic
ADHOC_CHAR_VAL	Adhoc Characteristic Value	Price List Characteristic
CHAR_VAL	Characteristic Value	Price List Characteristic
PRICELIST_ID	Price List ID	Price List
PL_TYPE	Price List Type	Price List
PL_STATUS_FLG	Status	Price List
SET_OPERATION	This is a BOOLEAN value. If **TRUE** - Reset operation configured at service level. If **FALSE** - Validate the fetch function output i.e. Throws error if returns NULL or multiple records. If **NULL** – Will do nothing but only throw an error if function return multiple records.	

List of output field/column values:

Output Parameter	Description	Table Description
PRICELIST_ID	Price List ID	Price List
PL_TYPE	Price List Type	Price List
PL_STATUS_FLG	Status	Price List
PL_START_DT	Effective Start Date	Price List
PL_END_DT	Effective End Date	Price List
PL_GLOBAL_SW	Global Price List Switch	Price List
CHAR_TYPE_CD	Characteristic Type Code	Price List Characteristics
CHAR_VAL	Characteristic Value	Price List Characteristics
EFFDT	Effective Date	Price List Characteristics
CHAR_VAL_FK1	Characteristic Value FK1	Price List Characteristics
CHAR_VAL_FK2	Characteristic Value FK2	Price List Characteristics
CHAR_VAL_FK3	Characteristic Value FK3	Price List Characteristics
CHAR_VAL_FK4	Characteristic Value FK4	Price List Characteristics
CHAR_VAL_FK5	Characteristic Value FK5	Price List Characteristics
VERSION	Version	Price List

Copyright © 2024, Oracle and/or its affiliates.

Fetch Rate Map Id

These function can be used to get and map the Rate Component details with its other corresponding column/output values listed below.

Either combination of input parameters **RS_CD, RC_DESCR, RS_DESCR, RC_SEQ** can be passed or can directly pass input **RC_MAP_ID** to this function.

Function:

```
FETCH(RATE_COMPONENT[COLUMN_NAME,RS_CD,RC_DESCR,RS_DESCR,RC_SEQ,RC_MAP_ID,
SET_OPERATION])
```

Expected list of input field/column values:

Input Parameter	Description	Table Description
COLUMN_NAME	Column name to be mapped from the returned RESULTSET. i.e. **OUTPUT** field name	
RS_CD	Rate Schedule	Rate Component Map
RC_DESCR	Rate Component Description	Rate Component Language
RS_DESCR	Rate Schedule Description	Rate Schedule Language
RC_SEQ	Sequence	Rate Component Map
RC_MAP_ID	RC Map ID	Rate Component Map
SET_OPERATION	This is a BOOLEAN value. If **TRUE** - Reset operation configured at service level. If **FALSE** - Validate the fetch function output i.e. Throws error if returns NULL or multiple records. If **NULL** – Will do nothing but only throw an error if function return multiple records.	

List of output field/column values:

Output Parameter	Description	Table Description
RS_CD	Rate Schedule	Rate Component Map
RC_DESCR	Rate Component Description	Rate Component Language
RS_DESCR	Rate Schedule Description	Rate Schedule Language
RC_SEQ	Sequence	Rate Component Map
RC_MAP_ID	RC Map ID	Rate Component Map

Fetch User

These function can be used to get and map the User details with its other corresponding column/output values listed below.

Input parameter **USER_ID** is mandatory to invoke this function.

Copyright © 2024, Oracle and/or its affiliates.

Function:

```
FETCH(USER[COLUMN_NAME,USER_ID,EXT_USER_ID,USER_TYPE_FLG,USER_ENABLE_FLG,
OWNER_FLG,SET_OPERATION])
```

Expected list of input field/column values:

Input Parameter	Description	Table Description
COLUMN_NAME	Column name to be mapped from the returned RESULTSET. i.e. **OUTPUT** field name	
USER_ID	User ID	User
EXT_USER_ID	External User Id	User
USER_TYPE_FLG	User Type	User
USER_ENABLE_FLG	User Enable	User
OWNER_FLG	Owner	User
SET_OPERATION	This is a BOOLEAN value. If **TRUE** - Reset operation configured at service level. If **FALSE** - Validate the fetch function output i.e. Throws error if returns NULL or multiple records. If **NULL** – Will do nothing but only throw an error if function return multiple records.	

List of output field/column values:

Output Parameter	Description	Table Description
USER_ID	User ID	User
FIRST_NAME	First Name	User
LAST_NAME	Last Name	User
TNDR_SOURCE_CD	Tender Source	User
DISP_PROF_CD	Display Profile Code	User
TD_ENTRY_AGE_DAYS1	Lower Age Limit for Yellow Bar	User
TD_ENTRY_AGE_DAYS2	Upper Age Limit for Yellow Bar	User
ACCESS_GRP_CD	Access Group	User
EMAILID	Email Address	User
USER_TYPE_FLG	User Type	User
DISP_ALL_PREM_SW	Display All Premises	User
OWNER_FLG	Owner	User
EXT_USER_ID	External User ID	User

Copyright © 2024, Oracle and/or its affiliates.

TIME_ZONE_CD	Time Zone	User
USER_ENABLE_FLG	User Enable	User
VERSION	Version	User

Check Access Group

These function can be used to validate the input Access Group.

Input parameter **USER_ID** is mandatory to invoke this function.

Function:

```
FETCH(ACCESS_GRP[ACCESS_GRP])
```

Expected list of input field/column values:

Input Parameter	Description	Table Description
ACCESS_GRP	Access Group	Access Group

List of output field/column values:

Output Parameter	Description	Table Description
ACCESS_GRP	Access Group	Access Group

Copy Functions

Copy function can be used to get the required column value of any existing entity in ORMB system. This function will invoke the Business Object (BO) or Business Service (BS) that is provided as parameter value i.e. "**SERVICE_NAME**" and get the required column or list block i.e. "**OUTPUT_XPATH**" from its returned XML data payload.

COPY functions can be used in only **FIELD TRANSFORMATION** section but not in HEADER and FOOTER TRANSFORMATION.

COPY with **BO** – To invoke BO, parameter **SERVICE_TYPE** value must be **BO.** This function will require an input of primary key of a primary table defined in its corresponding Maintenance Object (MO) meta-.

COPY with **BS** – To invoke BS, parameter **SERVICE_TYPE** value must be **BS.** This function can have any of those possible inputs defined in BS HEADER schema.

Function:

```
COPY(SERVICE_TYPE, SERVICE_NAME, OUTPUT_XPATH, SET_OPERATION,
INPUT[ELEMENT_XPATH=INPUT_VALUE], PREPROCESS[OPERATION], -
POSTPROCESS[OPERATION],
FILTER[ELEMENT_XPATH=INPUT_VALUE], UPD[ELEMENT_XPATH=INPUT_VALUE])
```

INPUT Parameter details:

SERVICE_TYPE

Type of Service to be invoked. It can have either of this two values "BO" or "BS".

SERVICE_NAME

ORMB "BO" or "BS" service names that is to be invoked for copying.

OUTPUT_XPATH

Copyright © 2024, Oracle and/or its affiliates.

COPY function will be getting the response in XML form.So, this will be theXPATH of that element whose value we want to map from the response XML. This XPATH can be either of an individual element or the list block in the response XML.

List Block XPATH will allow to copy the complete list block from the response XML to the input payload.

Note: For list block copy we have to suffix the XPATH with "**/list**" in the corresponding 'MAP FIELD XPATH' column value.

Example:

```
 COPY(BO,C1-AccountBO,C1-AccountBO/accountAutopay,
FILTER[C1-AccountBO/accountAutopay/endDate=''],UPD[AUTO PAY EXPIRES '
ON= -'2020-12-31',
C1-AccountBO/accountAutopay/endDate=AUTO PAY EXPIRY])
```

In this above COPY function example, we are mapping the "C1-AccountBO/accountAutopay" list block from the copied entity to the input service payload.

COPY for list block retrieval will be invoked with having service payload XML as an input. This service payload must have the required primary key inputs for the BO to be invoked.

If we are invoking BO '**C1-AccountBO**', then, this input service payload must have '**C1-AccountBO/accountId'** element value i.e. Primary key value of primary table defined in Account MO.

SET_OPERATION

This can have a **BOOLEAN** value.

If **TRUE** - Reset operation configured at service level.

If **FALSE** - Validate the fetch function output i.e. Throws error if returns NULL or multiple records.

If **NULL** – Will do nothing but only throw an error if function return multiple records.

INPUT [ELEMENT_XPATH=INPUT_VALUE]

This is an optional input parameter block. Applicable only if we want to get the individual element value i.e. not the list block from response XML.

This block can have multiple comma-separated inputs.

- **ELEMENT_XPATH –** XPATH of the input element i.e. key XPATH in the BO/BS to be invoked
- **INPUT_VALUE –** Input element i.e. key value. This will be used to invoke theBO/BS

Input values can have LITERAL values i.e. static values enclosed in single quotes or the CONSTANT values i.e. ":BUS_DATE;BUS_DTTM;STD_DATE;STD_DTTM;SYSDATE;SYS_DTTM;CHAR_DATE;FILENAME;FILE_ID" or those configured FIELD names in DATA TRANSFORMATION corresponding to its section i.e. HEADER or FOOTER or FIELD TRANSFORMATION whose mapped value will be referred.

Example:

COPY(BO,CM-TaxRates,CM-TaxRates/lookupValue,true,**INPUT[CM-TaxRates/bo=BUS_OBJ_CD,CM-TaxRates/lookupValue=LOOKUPKEY]**)

PREPROCESS [OPERATION]

This is an optional parameter block. This can be used to perform the required pre-processing and then processing the actual record.

- **OPERATION –** This is optional parameter. It can have NULL value. If NULL, then, it will have the operation value that is configured in FRT at service level.
- **INPUT** block is mandatory for **PREPROCESS** and **POSTPROCESS.** Multiple records can be added or updated using FETCH function in the INPUT block. Same applies to **POSTPROCESS.**

Copyright © 2024, Oracle and/or its affiliates.

Example: COPY(BO,C1_PERSON_BO,C1_PERSON_BO/personPerson,**INPUT[PARENT_PER_ID=FETCH(PER_RELATION[PER_ID1,null,CHILD_PER_ID,null,null,:BUS_DATE,** && C1_PERSON_BO/personPerson/endDate=" && C1_PERSON_BO/personPerson/personId1!=PARENT_PER_ID || CHILD_PER_ID=CHILD_PER_ID && C1_PERSON_BO/personPerson/endDate.ISAFTER(:BUS_DATE) && C1_PERSON_BO/personPerson/personId1!=PARENT_PER_ID],UPD[C1_PERSON_BO/personPerson/endDate=END_DATE_UPD])

POSTPROCESS [OPERATION]

It is an optional parameter block. This can be used to perform the required post-processing and then processing the actual record.

- **OPERATION –** This is optional parameter. It can have NULL value. If NULL, then, it will have the operation value that is configured in FRT at service level.

FILTER [ELEMENT_XPATH=INPUT_VALUE]

It is an optional parameter block. These can be used if we want to apply filter and get only the specific element value from the possible list of blocks in response XML.

- **ELEMENT_XPATH** - XPATH of the filter element i.e. filter key XPATH in the response XML.
- **INPUT_VALUE** - Filter condition value i.e. key value. This will be used to filter the value from response XML.

Input values can have LITERAL values i.e. static values enclosed in single quotes or the CONSTANT values i.e. ":BUS_DATE;BUS_DTTM;STD_DATE;STD_DTTM;SYSDATE;SYS_DTTM;CHAR_DATE;FILENAME;FILE_ID" or those configured FIELD names in DATA TRANSFORMATION corresponding to its section i.e. HEADER or FOOTER or FIELD TRANSFORMATION whose mapped value will be referred.

Filter criteria can be defined using an expression using supported conditional and logical operators.

Example: COPY(BO,C1_PERSON_BO,C1_PERSON_BO/personPerson,INPUT[PARENT_PER_ID=FETCH(PER_RELATION[PER_ID1,null,CHILD_PER_ID,null,null,:BUS_DATE,false]) **FILTER** [CHILD_PER_ID=CHILD_PER_ID && C1_PERSON_BO/personPerson/endDate=" && C1_PERSON_BO/personPerson/personId1!=PARENT_PER_ID || CHILD_PER_ID=CHILD_PER_ID && C1_PERSON_BO/personPerson/endDate.ISAFTER(:BUS_DATE) && C1_PERSON_BO/personPerson/personId1!=PARENT_PER_ID],UPD[C1_PERSON_BO/personPerson/endDate=END_DATE_UPD])

Equality, Relational, and Conditional Operators

The equality and relational operators determine if one operand is greater than, less than, equal to, or not equal to another operand.

List of Operators:

Operator	**Description**	**Supported Data Type**
=	Equal to	CHAR, NUMBER
!=	Not Equal to	CHAR, NUMBER
<	Less than	NUMBER
<=	Less than or Equal to	NUMBER
>	Greater than	NUMBER
>=	Greater than or Equal to	NUMBER
ISBEFORE	Less than	DATE
ISBEFOREOREQUAL	Less than or Equal to	DATE
ISAFTER	Greater than	DATE
ISAFTEROREQUAL	Greater than or Equal to	DATE

Copyright © 2024, Oracle and/or its affiliates.

Operator	Description	Supported Data Type
ISEQUAL	Equal to	DATE
ISNOTEQUAL	Not Equal to	DATE

Logical Operators:

The && and || operators perform Conditional-AND and Conditional-OR operations on two boolean expressions. These operators exhibit "short-circuiting" behavior, which means that the second operand is evaluated only if needed.

Note: No enclosing brackets allowed.

List of Logical Operators:

Operator	Description
&&	AND
||	OR

UPD[ELEMENT_XPATH=INPUT_VALUE]

This is an optional input parameter. These can be used to update the element value in the copied XML block.

Multiple element values can be updated in a copied XML block having the multiple comma-separated element key values.

- **ELEMENT_XPATH** - XPATH of the element in the copied XML response for which the value is to be updated.
- **INPUT_VALUE** - Input element i.e. key value. This will be used to update the element value in the copied XML block.

Input values can have LITERAL values i.e. static values enclosed in single quotes or the CONSTANT values i.e. ":BUS_DATE,:BUS_DTTM,:SYSDATE,:SYS_DTTM,:CHAR_DATE,:FILENAME,:FILE_ID,:REMOVE,:REMOVE_FLD" or those configured FIELD names in DATA TRANSFORMATION corresponding to its section i.e. HEADER or FOOTER or FIELD TRANSFORMATION whose mapped value will be referred.

Example: COPY(BO,C1-AccountBO,C1-AccountBO/accountAutopay,FILTER[C1-AccountBO/accountAutopay/endDate=''],**UPD**[AUTO PAY EXPIRES ON= '2020-12-31',C1-AccountBO/accountAutopay/endDate=AUTO PAY EXPIRY])

String Handling Functions

String handling functions can be used for modifying the individual field inputs before formation of the input payload. These functions can also be used for validations in the SCRIPT function.

STRING functions can be used in all three sections of DATA TRANSFORMATION i.e. HEADER, FOOTER or FIELD.

String functions can be used in two ways:

- **Modify individual field input** –

These can be done only for the input file data i.e. not for DEFAULT values with ‘SEQUENCE - 0’. Fields with SEQUENCE ‘0’ can use this String functions only with SCRIPT function.

EDIT INPUT flag should be selected to have default values configured for FIELD’s that have SEQUENCE not equal to ZERO i.e. ‘0’.

- **Use with SCRIPT function** -

SCRIPT function with STRING functions use has no constraints. Invoking STRING function with in SCRIPT has a different SYNTAX. STRING functions are invoked with reference to the FIELD NAME i.e. **FIELD_NAME.STRING_FUNCTION**

Copyright © 2024, Oracle and/or its affiliates.

Example:

```
SCRIPT([PERSONIDNUMBER!=''&&PERSONIDTYPE=''],{PERSONIDNUMBER.SUBSTR(0,3)})
```

List of functions:

- Substring
  - Substring from search string(Inclusive) - **SUBSTR('startString')**
  - Substring from startIndex - **SUBSTR(startIndex)**
  - Substring from search string(Inclusive) to endIndex - **SUBSTR('startString',endIndex)**
  - Substring from startIndex to search string(Inclusive) - **SUBSTR(startIndex,'endString')**
  - Substring from startIndex to endIndex - **SUBSTR(startIndex,endIndex)**
  - Substring between given string inputs - **SUBSTR('startString','endString')**
  - Substring to get last index Char Sequence - **SUBSTRLAST(startIndex)**
- Prefix the input string - **PREFIX('prefixString')**
- Suffix the input string - **SUFFIX('suffixString')**
- Convert string to uppercase - **UPPERCASE()**
- Convert string to lowercase - **LOWERCASE()**
- Right trim the input string - **RTRIM()**
- Left trim the input string - **LTRIM()**
- Right and Left trim the input string - **TRIM()**
- Concat - **CONCAT(FLD_NAME1||FLD_NAME2)**
- **ISNUMBER(INPUT_VALUE)** – This can be used only with SCRIPT function
- **LENGTH(INPUT_VALUE)** – This can be used only with SCRIPT function
- **ISUTF8 (INPUT_VALUE)** – This can be used only with SCRIPT function
- **CONTAINS (INPUT_VALUE,SEARCH_STRING,EXPECTED_OUTPUT)** – This can be used only with SCRIPT function
- **STARTSWITH(INPUT_FLD,SEARCH_STRING,EXPECTED_OUTPUT)–** This can be used only with SCRIPT function
- **ENDSWITH(INPUT_FLD,SEARCH_STRING,EXPECTED_OUTPUT)**– This can be used only with SCRIPT function

Substring

Returns a string that is a substring of the corresponding field input string.

Substring from search string

The substring begins from the characters that are specified as **startString** and extends to the end of this string.

Function: SUBSTR('startString')

Examples:

"unhappy" - SUBSTR(‘ha’) returns "happy"

"Harbison" - SUBSTR(‘rbi’) returns "rbison"

Parameters:

startString the starting string, inclusive. This input string should be enclosed in single quotes.

Returns:

The specified substring.

Substring from startIndex

The substring begins from the given startIndex and extends to the end of this string.

Function: SUBSTR(startIndex)

Copyright © 2024, Oracle and/or its affiliates.

Examples:

"unhappy" - SUBSTR(2) returns "happy"

"Harbison" - SUBSTR(3) returns "bison"

Parameters:

startIndex the starting index, inclusive.

Returns:

The specified substring.

Substring from search string to endIndex

The substring begins from the character that are specified as **startString** and extends to the character at index **endIndex** – 1.

Function: SUBSTR('startString',endIndex)

Examples:

"unhappy" - SUBSTR(‘ha’,5) returns "hap"

Parameters:

startString the starting string, inclusive. This input string should be enclosed in single quotes.

endIndex the end index, exclusive.

Returns:

The specified substring.

Substring from startIndex to search string

The substring begins from the given **startIndex** and extends till the given **endString** of this string.

Function: SUBSTR(startIndex,'endString')

Examples:

"unhappy" - SUBSTR(2, 'ppy') returns "happy"

"Harbison" - SUBSTR(3,‘so’) returns "rbiso"

Parameters:

startIndex the end index, inclusive.

endString the end string, inclusive. This input string should be enclosed in single quotes.

Returns:

The specified substring.

Substring from startIndex to endIndex

The substring begins at the specified **startIndex** and extends to the character at index **endIndex** - 1. Thus the length of the substring is **endIndex**- **startIndex**.

Function: SUBSTR(startIndex, endIndex)

Examples:

```
-"hamburger" -- SUBSTR(4, 8) returns -"urge"
```

```
-"smiles" -- SUBSTR(1, 5) returns -"mile"
```

Parameters:

Copyright © 2024, Oracle and/or its affiliates.

startIndex the starting index, inclusive.

endIndex the ending index, exclusive.

Returns:

The specified substring.

Substring between given string inputs

The substring returns the string **between** given **startString** and **endString** i.e. String exclusive of the given string inputs.

Function: SUBSTR('startString', 'endString')

Examples:

"beautiful" - SUBSTR(‘ea’,’fu’) returns "uti"

Parameters:

startString the starting string, exclusive. This input string should be enclosed in single quotes.

endString the end string, exclusive. This input string should be enclosed in single quotes.

Returns:

The specified substring.

Substring to get last index Char Sequence

The substring returns this no. of last char sequence i.e. given **length_of_char_sequence** for the input string

Function: SUBSTRLAST (length_of_char_sequence)

Examples:

"beautiful" - SUBSTRLAST(5) returns "tiful"

Parameters:

length_of_char_sequence is the no. of char sequence from the end of String.

Returns:

The specified substring.

Prefix the input string

The string is prefixed with the characters in the specified **prefixString**.

Function: PREFIX('prefixString')

Examples:

"morning" - PREFIX (‘this’) returns " this morning"

Parameters:

prefixString the prefix String. This input string should be enclosed in single quotes.

Substring from search string

The string is suffixed with the characters in the specified **suffixString**.

Function: SUFFIX('suffixString')

Examples:

"morning" - SUFFIX(‘tea’) returns "morning tea"

Parameters:

suffixString the suffix string. This input string should be enclosed in single quotes.

Copyright © 2024, Oracle and/or its affiliates.

Convert string to uppercase

Converts all of the characters in this String to upper case using the rules of the default locale.

Function: UPPERCASE ()

Examples:

"happy" - UPPERCASE() returns "HAPPY"

Returns:

The uppercase string.

Convert string to lowercase

Converts all of the characters in this String to lower case using the rules of the default locale.

Function: LOWERCASE ()

Examples:

"HappyBirthday" - LOWERCASE() returns "happybirthday"

Returns:

The lowercase string.

Right trim the input string

Returns a string whose value is this string, with any trailing whitespace removed.

This method may be used to trim whitespace from the end of a string.

Function: RTRIM()

Examples:

" unhappy " – RTRIM() returns " unhappy"

Returns:

A string whose value is this string, with any trailing white space removed, or this string if it has no trailing white space.

Left trim the input string

Returns a string whose value is this string, with any leading whitespace removed.

This method may be used to trim whitespace from the start of a string.

Function: LTRIM()

Examples:

" unhappy " – LTRIM() returns "unhappy "

Returns:

A string whose value is this string, with any leading white space removed, or this string if it has no leading white space.

Trim the input string

Returns a string whose value is this string, with any leading and trailing whitespace removed.

This method may be used to trim whitespace from the beginning and end of a string.

Function: TRIM()

Examples:

" unhappy " – TRIM() returns "unhappy"

Copyright © 2024, Oracle and/or its affiliates.

Returns:

A string whose value is this string, with any leading and trailing white space removed, or this string if it has no leading or trailing white space.

ISNUMBER

Returns true if and only if this string contains only numeric values. This function can be used only with **SCRIPT** function.

Function: ISNUMBER(INPUT_VALUE)

Examples:

- If **PERSONIDNUMBER** = ‘2020-04-30’

SCRIPT([PERSONIDNUMBER!=" && **ISNUMBER(PERSONIDNUMBER)**],{PERSONIDNUMBER;”})

Here, SCRIPT function returns **‘’ i.e. EMPTY_STRING**

- If **PERSONIDNUMBER** = ‘20200430’

SCRIPT([PERSONIDNUMBER!=" && **ISNUMBER(PERSONIDNUMBER)**],{PERSONIDNUMBER;”})

Here, SCRIPT function returns **PERSONIDNUMBER**

Parameters:

INPUT_VALUE the sequence to search for

Returns:

TRUE if this string contains only numeric values, **FALSE** otherwise

LENGTH

This can be used to compare the string length. It has to be used with conditional operators and numeric value. This function can be used only with **SCRIPT** function.

Function: LENGTH(INPUT_VALUE)

Examples:

- If **PERSONIDNUMBER** = ‘2020-04-30’

SCRIPT([PERSONIDNUMBER!=" && **LENGTH(PERSONIDNUMBER)>15**],{PERSONIDNUMBER;”}

Here, SCRIPT function returns **‘’ i.e. EMPTY_STRING**

- If **PERSONIDNUMBER** = ‘20200430’

SCRIPT([PERSONIDNUMBER!=" && **LENGTH(PERSONIDNUMBER)=8**],{PERSONIDNUMBER;”})

Here, SCRIPT function returns **PERSONIDNUMBER**

Parameters:

INPUT_VALUE the LENGTH to compare with

Returns:

TRUE if this LENGTH condition satisfies, **FALSE** otherwise

ISUTF8

Returns true if and only if this string contains only UTF8 character set. This function can be used only with **SCRIPT** function.

Function: ISUTF8(INPUT_VALUE)

Parameters:

Copyright © 2024, Oracle and/or its affiliates.

INPUT_VALUE the character set to check for encoding

Returns:

TRUE if this string contains the string with UTF8 character set, **FALSE** otherwise.

CONTAINS

Returns true if and only if this string contains the specified sequence of char values. This function can be used only with **SCRIPT** function.

Function: CONTAINS(INPUT_FLD,SEARCH_STRING,EXPECTED_OUTPUT)

Examples:

- If **PERSONIDNUMBER** = ‘2020-04-30’

SCRIPT([PERSONIDNUMBER!="&&CONTAINS(**PERSONIDNUMBER**,‘2004’,TRUE)], {PERSONIDNUMBER;’’})

Here, SCRIPT function returns **‘’ i.e. EMPTY_STRING**

- If **PERSONIDNUMBER** = ‘20200430’

SCRIPT([PERSONIDNUMBER!="&&CONTAINS(**PERSONIDNUMBER,**‘2004’,TRUE)], {PERSONIDNUMBER;’’})

Here, SCRIPT function returns **PERSONIDNUMBER**

Parameters:

INPUT_FLD the input field for whose value we will have a search

SEARCH_STRING the sequence to search for

EXPECTED_OUTPUT the expected result. This can be TRUE or FALSE.

Returns:

TRUE if this returns the **EXPECTED_OUTPUT**, **FALSE** otherwise.

STARTSWITH

Returns true if and only if this string starts with the specified sequence of char values. This function can be used only with **SCRIPT** function.

Function: STARTSWITH(INPUT_FLD,SEARCH_STRING,EXPECTED_OUTPUT)

Examples:

- If **PERSONIDNUMBER** = ‘2020-04-30’

SCRIPT([PERSONIDNUMBER!="&&STARTSWITH(**PERSONIDNUMBER**,‘202004’,TRUE)], {PERSONIDNUMBER;’’})

Here, SCRIPT function returns **‘’ i.e. EMPTY_STRING**

- If **PERSONIDNUMBER** = ‘20200430’

SCRIPT([PERSONIDNUMBER!="&&STARTSWITH(**PERSONIDNUMBER**‘202004’,TRUE)], {PERSONIDNUMBER;’’})

Here, SCRIPT function returns **PERSONIDNUMBER**

Parameters:

INPUT_FLD the input field value to be checked for

SEARCH_STRING the sequence to search for

Copyright © 2024, Oracle and/or its affiliates.

EXPECTED_OUTPUT the expected result. This can be TRUE or FALSE.

Returns:

TRUE if this returns the **EXPECTED_OUTPUT**, **FALSE** otherwise.

ENDSWITH

It returns true if and only if this string ends with the specified sequence of char values. This function can be used only with **SCRIPT** function.

Function: ENDSWITH(INPUT_FLD,SEARCH_STRING,EXPECTED_OUTPUT)

Examples:

- If **PERSONIDNUMBER** = ‘2020-04-30’

SCRIPT([PERSONIDNUMBER!="&&ENDSWITH(**PERSONIDNUMBER**,’0430’,TRUE)], {PERSONIDNUMBER;’’})

Here, SCRIPT function returns **‘’ i.e. EMPTY_STRING**

- If **PERSONIDNUMBER** = ‘20200430’

SCRIPT([PERSONIDNUMBER!="&&STARTSWITH(**PERSONIDNUMBER**‘0430’,TRUE)], {PERSONIDNUMBER;’’})

Here, SCRIPT function returns **PERSONIDNUMBER**

Parameters:

INPUT_FLD the input field value to be checked for

SEARCH_STRING the sequence to search for

EXPECTED_OUTPUT the expected result. This can be TRUE or FALSE.

Returns:

TRUE if this returns the **EXPECTED_OUTPUT**, **FALSE** otherwise

Script Functions

The **SCRIPT function** in ORMB is used for **conditional execution and data transformation** in file uploads. It enables users to define **control statements** and **execution blocks** for mapping, validation, and transformation logic.

SCRIPT Function Structure

The **SCRIPT function** consists of two blocks:

Block Type	Description
Control Statement Block `[ … ]`	Used for decision-making. Conditions within this block are evaluated from **left to right**.
Execution Block `{ … }`	Defines the actions to be performed when conditions in the control block are met.

SCRIPT Function

Syntax

```
SCRIPT([Control_Condition],{Execution_Action})
```

Copyright © 2024, Oracle and/or its affiliates.

Example

```
SCRIPT([EFFECTIVEDATE.isAfter(:CHAR_DATE)], -
{ERROR(90003,282,PERSONIDNUMBER,:CHAR_DATE,'Validation Error')})  -
```

Operators in SCRIPT Function:Equality, Relational, and Conditional Operators determine whether one operand is **greater than, less than, equal to, or not equal** to another:

Operator	Description	Supported Data Type
=	Equal to	CHAR, NUMBER
!=	Not Equal to	CHAR, NUMBER
<	Less than	NUMBER
<=	Less than or Equal to	NUMBER
>	Greater than	NUMBER
>=	Greater than or Equal to	NUMBER
ISBEFORE	Less than	DATE
ISBEFOREOREQUAL	Less than or Equal to	DATE
ISAFTER	Greater than	DATE
ISAFTEROREQUAL	Greater than or Equal to	DATE
ISEQUAL	Equal to	DATE
ISNOTEQUAL	Not Equal to	DATE

Example:

This function throws an **error** if `EFFECTIVEDATE` is later than `CHAR_DATE`.

EVAL function

This function can be used to evaluate the mathematical expression. It supports expressions that are supported in JavaScript. For evaluating expression we are using JavaScript engine.

EVAL functions can be used in all three sections of DATA TRANSFORMATION i.e. HEADER, FOOTER or FIELD.

Function:

```
 EVAL([EXPRESSION];INPUT[PARAM_LIST_USED_IN_EXPRESSION])
```

INPUT parameters has two sections separated by semicolon (**;**) delimiter.

- **EXPRESSION** – This will be the expression that we want to evaluate. It is enclosed with square brackets i.e. (**[]**)

```
EVAL([(FLD1+FLD2+10)/2];INPUT[FLD1,FLD2])
```

- **PARAM_LIST_USED_IN_EXPRESSION** – This will be the list of Field Names (only those defined in Field Transformation) used in expression. It is enclosed with square brackets i.e. **[]**

```
EVAL([(FLD1+FLD2+10)/2];INPUT[FLD1,FLD2])
```

If FLD1 = 5 and FLD2=7 then this function will return 11

Copyright © 2024, Oracle and/or its affiliates.

DUPLICATE function

This function can be used to create the duplicate record of only those required processed record in a file with also having the updates in that record to be duplicated. This function has to be always used with SCRIPT function. In SCRIPT function this can be used only in Execution Block i.e. {…}

DUPLICATE functions can be used only in Field Transformation i.e. not in HEADER or FOOTER.

Example:

```
SCRIPT([ACCT_ID_1!=''],{DUPLICATE(ZACCT_ID_CUST=ZCASHACCT_ID_UPD,
ZACCTCHAR_CUST_0_UPD=ZCASHACCT_ID_UPD,ZACCTCHAR_CUST_1_UPD=ZCASHACCT_ID_UPD,
ZACCTCHAR_CUST_2_UPD=ZCASHACCT_ID_UPD,ZACCTCHAR_CUST_3_UPD=ZCASHACCT_ID_UPD);
‘123456789’})
```

SCRIPT workflow: IF the **ACCT_ID_1** is not empty THEN execute **DUPLICATE** function and map the returned value. ELSE **123456789**

Function:

```
TODO(TODO_TYPE,TODO_ROLE,ASSIGN_USER,STATUS,SINGLE_TODO,SORT_KEY[KEY_LIST],
DRILL_KEY[KEY_LIST],CHAR[TYPE=VALUE],MSG[MSG_CAT,MSG_NUM,parm1,param2..])
```

INPUT Parameter – This can have list of COMMA-SEPARATED KEY-VALUE pairs that is to be updated in duplicated record payload.

- **KEY** – This is the XPATH of element in the record payload for which the value is to be updated. It can be the FIELD NAME configured in FIELD TRANSFORMATION whose corresponding XPATH will be referred or we can directly have the XPATH as a KEY.
- **VALUE** – It can have the LITERAL values i.e. static values enclosed in single quotes or the CONSTANT values i.e. ”:BUS_DATE, :BUS_DTTM , :STD_DATE, :STD_DTTM, :SYSDATE, :SYS_DTTM, :CHAR_DATE, :FILENAME, :FILE_ID“ or those configured FIELD names in FIELD TRANSFORMATION whose mapped value will be referred.

Config Function

This function can be used to get the feature configuration value.

CONFIG function can be used in all three sections of DATA TRANSFORMATION i.e. HEADER, FOOTER or FIELD.

Function:

```
CONFIG(featureConfigName, optionType, sequence)
```

- **featureConfigName** – Feature configuration name defined in ORMB system
- **optionType** – Option Type in feature configuration
- **sequence** – Sequence Number in feature configuration

INPUT parameters can have **Literal** values i.e. static values enclosed in single quotes or those configured Field names in Data Transformation whose mapped value will be referred.

Header Function

This function can be used to get the HEADER value referred in FIELD TRANSFORMATION section. HEADER function can be used only in FIELD TRANSFORMATION i.e. not in HEADER, FOOTER.

Function:

```
HEADER(INPUT_HEADER_FIELD_NAME)
```

Copyright © 2024, Oracle and/or its affiliates.

INPUT_HEADER_FIELD_NAME - Only FIELD names configured in HEADER TRANSFORMATION can be referred.

Example :

```
HEADER(TXNSOURCECD)
```

TO DO Generation

This function can be used to generate **TO DO** for failure or success of each individual record execution status or a single **TO DO** after completion of batch to get the summary of all the failures or multiple TODO's for each of the selected failures in this particular execution.

TODO functions can be used in all three sections of **DATA TRANSFORMATION** i.e. HEADER, FOOTER or FIELD.

Function:

```
TODO(TODO_TYPE,TODO_ROLE,ASSIGN_USER,STATUS,SINGLE_TODO,SORT_KEY[KEY_LIST],
DRILL_KEY[KEY_LIST],CHAR[TYPE=VALUE],MSG[MSG_CAT,MSG_NUM,parm1,param2..])
```

TO DO generation is allowed at two levels,

- **RECORD LEVEL**: Parameter value of '**STATUS'** should not be **NULL**. It must be either **TRUE** or **FALSE.**
  - **TODO** for **FAILURE**: Parameter value of '**STATUS'** should be **FALSE**
  - **TODO** for **SUCCESS**: Parameter value of '**STATUS'** should be **TRUE**
- **BATCH LEVEL**: Parameter value of '**STATUS'** should be **NULL**.

INPUT Parameters:

- **TODO_TYPE** – TODO Type for which TODO is to be created
- **TODO_ROLE** – TODO Role will be responsible to work on this TODO
- **ASSIGN_USER** – Created TODO will be assigned to this USER
- **STATUS** – Possible values are TRUE, FALSE or NULL.
  - If **TRUE –** TODO will be created after **successful** execution of each record.
  - If **FALSE –** TODO will be created after **failed** execution of each record.
  - If **NULL –** TODO will be created after success or failed execution of this batch.
- **SINGLE_TODO –** This parameter will be used to decide if this TODO is applicable for all the defined SERVICES in this FRT. Possible values are **TRUE** or **FALSE**. Parameter in use if FRT has multiple SERVICES configured.
  - If **TRUE** – This TODO function will be applicable for TODO generation of all the SERVICES in this FRT
  - If **FALSE** – This TODO function will be applicable for TODO generation in only that SERVICE mentioned in **MAP FILED XPATH** value against which this function is configured.
- **SORT_KEY[KEY_LIST]** – Sort Key required for the TODO TYPE. Multiple keys i.e. COMMA-SEPARATED can be passed. This is optional parameter block.

Example:

```
TODO('C1FUDFLT','F1-DFLT',null,NULL,TRUE,SORT_KEY[:FILE_ID],
DRILL_KEY[:FILE_ID,:FILENAME],CHAR['ZZACCT-E'='9901765140'],
MSG[90003,302,:FILE_ID,:FILENAME,:BATCH_CD,:BATCH_NUM])
```

- **DRILL_KEY[KEY_LIST]** – Drill Key required for the TODO TYPE. Multiple keys i.e. COMMA-SEPARATED can be passed. This is optional parameter block.

Example:

```
TODO('C1FUDFLT','F1-DFLT',null,NULL,TRUE,SORT_KEY[:FILE_ID],
DRILL_KEY[:FILE_ID,:FILENAME],CHAR['ZZACCT-E'='9901765140'],
```

Copyright © 2024, Oracle and/or its affiliates.

```
MSG[90003,302,:FILE_ID,:FILENAME,:BATCH_CD,:BATCH_NUM])
```

- **MSG[MSG_CAT,MSG_NUM,PARAM_LIST]–** Message for which the TODO generation should be done. Multiple keys i.e. COMMA-SEPARATED can be passed. This is optional parameter block.

INPUT parameters – These are the MESSAGE meta-data defined in ORMB.

MSG_CAT – Message Category

MSG_NUM – Message Number

PARAM_LIST – Input parameter list (COMMA-SEPARATED) for this message

Example:

```
TODO('C1FUDFLT','F1-DFLT',null,NULL,TRUE,SORT_KEY[:FILE_ID],
DRILL_KEY[:FILE_ID,:FILENAME],CHAR['ZZACCT-E'='9901765140'],
MSG[90003,302,:FILE_ID,:FILENAME,:BATCH_CD,:BATCH_NUM])
```

TODO generation has two behaviors for MSG block in record failures,

- TODO function **without** MSG Block
  - **Record Level** - TODO will be created after every **failed** execution of each record.
  - **Batch Level -** Single TODO will be created after completion of batch with the summary of all the failure details.
- TODO function **with** MSG Block
  - **Record Level** - TODO will be created after failure for the mentioned Category and Number execution of each record.
  - **Batch Level -** Single TODO will be created after completion of batch only for the mentioned Category and Number.

Note: Input parameters can have Literal values i.e. **Static** values enclosed in single quotes or **Constant** values, i.e. “:BUS_DATE, :BUS_DTTM , :STD_DATE, :STD_DTTM, :SYSDATE, :SYS_DTTM, :CHAR_DATE, :FILENAME, :FILE_ID” or those configured Field names in Data Transformation whose mapped value will be referred.

Concat Function

The **CONCAT** function in Oracle Revenue Management and Billing (ORMB) is used to concatenate multiple **TRANSFORMATION FIELD** values or **LITERAL** values. Literal values must be enclosed in single quotes **(' ')**. The function allows concatenation across different sections of **DATA TRANSFORMATION**, including:

- **HEADER**
- **FOOTER**
- **FIELD TRANSFORMATION**

Function Syntax:

```
CONCAT(FLD_NAME1||FLD_NAME2||'LITERAL_VALUE')
```

Delimiter:

The **double pipe (||)** is the delimiter used to separate multiple input values within the CONCAT function.

Parameters:

- **FLD_NAME1, FLD_NAME2** – Any FIELD names from the DATA TRANSFORMATION section.
- **'LITERAL_VALUE'** – A fixed string enclosed in single quotes.

Return Value:

The CONCAT function returns a **string** representing the concatenated value of the input fields or literals.

Example:

Copyright © 2024, Oracle and/or its affiliates.

If the transformation field **TXNDTTM** has the value `'2020-04-30'`, the following function:

```
CONCAT(TXNDTTM -|| -'-00.00.00')  -
```

Returns:

`'2020-04-30-00.00.00'`

Field Transformation Configuration

The following configuration example demonstrates the usage of the CONCAT function in ORMB's **Field Transformation** section:

Field Name	Map Field XPath	Default Value	Date/Time Format
TXNDTTM	C1-TranDtlStageUpload/0/tranDtl/txnDttm	CONCAT(TXNDTTM	

Field Transformation - TXNDTTM

The image illustrates the Field Transformation configuration in ORMB, where the **TXNDTTM** field is mapped to **C1-TranDtlStageUpload/0/tranDtl/txnDttm**. The **Default Value** column demonstrates an example usage of the CONCAT function, concatenating **TXNDTTM** with **'-00.00.00'** to standardize the date-time format.

Note:

- CONCAT can be used in **FIELD TRANSFORMATION**, **HEADER**, and **FOOTER** sections.
- Ensure correct formatting of the concatenated values to match the expected data schema.

Configuring the Approval Workflow Process for File Upload Interface

The **Approval Workflow Process** for the **File Upload Interface** ensures robust governance, accountability, and transparency in processing uploaded files. By defining approval chains, group linkages, and associated settings, businesses can establish standardized procedures for data validation and ensure compliance with organizational policies. Implementing a structured approval workflow process for file uploads delivers the following benefits:

- **Improved Operational Efficiency** - It streamlines the approval process, reducing manual overhead.
- **Enhanced Accountability** - It clear roles and responsibilities minimize approval ambiguities.
- **Risk Mitigation** - It ensures that all files undergo thorough validation, reducing the likelihood of errors or non-compliance.
- **Scalability** - It supports growing organizational needs by allowing for the addition of new groups and chains without system disruptions.

Approval Workflow Group

An **Approval Workflow Group** defines a set of users or roles responsible for reviewing and approving files uploaded through the **File Upload Interface**. The key benefits are:

- **Centralized Management** - It allow businesses to organize approvers and their roles systematically.
- **Flexibility** - It can be configured to include multiple approver based on roles, departments, or project needs.
- **Compliance** - It ensures that the approval process adheres to regulatory and internal compliance guidelines.

For example an **Approval Workflow Group** for finance files may include approvers from both the finance and compliance departments to ensure thorough validation.

Copyright © 2024, Oracle and/or its affiliates.

To create approval workflow group refer to Defining an Approval Workflow Group.

For more information Refer Approval Workflow Group (Used for Searching)

Approval Workflow Chain

An **Approval Workflow Chain** defines the sequential order of approvals for a file or record. Each step in the chain is linked to a specific workflow group or role. As business perspective it streamlines the review process, ensuring each stakeholder is involved at the right stage and reduces the risk of incomplete or erroneous file uploads. It contains various features like:

- **Step-by-Step Validation** - Each step in the chain represents a layer of validation, ensuring that all necessary checks are performed.
- **Conditional Approvals** - It have specific conditions that can be attached to each step, such as budget thresholds or project requirements.

For example an uploaded file might first require approval from a **Supervisor**, followed by a **Compliance Officer**.

To create approval workflow chain refer to Defining an Approval Workflow Chain

For more information Refer Approval Workflow Chain (Used for Searching)

Approval Workflow Criterion Type

An **Approval Workflow Criterion Type** determine the conditions under which a file or record enters the approval workflow. These are configured based on file attributes, such as type or status. It enables customized routing rules that align with organizational needs, ensuring that the right people handle the right files. The common criterion types are:

- **File Type** - Approvals are routed based on the type of file being uploaded (For example financial, operational, or regulatory files).
- **Size Thresholds** - Large files may require additional approvals to ensure data integrity.
- **Priority Levels** - High-priority files are routed to senior approvers for expedited handling.

For example, configure criteria to trigger approval for files of type **Financial Transactions** or files larger than 1 GB.

To create approval workflow criterion type refer to Defining an Approval Workflow Criterion Type

For more information Refer Approval Workflow Criterion Type (Used for Searching)

Approval Workflow Group Chain Linkage

An **Approval Workflow Group Chain Linkage** connects **Approval Workflow Groups** with their respective chains to create a cohesive approval process. It ensures that approvals proceed through the appropriate sequence by linking defined groups to the configured chain. It have following advantages:

- **Scalability**- It allows for the inclusion of additional workflow groups without disrupting the existing setup.
- **Interdepartmental Collaboration**- It links groups from different departments, fostering collaboration and efficiency.

It is used to link compliance, legal, and operations groups in a workflow for regulatory filings to ensure all aspects of the document are thoroughly reviewed and approved. For example, link the **Finance Approvers** group to the **Finance Transaction Approval**.

To create approval workflow group chain linkage refer to Defining an Approval Workflow Group Chain Linkage

For more information Refer Approval Workflow Group Chain Linkage (Used for Searching)

Copyright © 2024, Oracle and/or its affiliates.

Approval Workflow Settings

Approval workflow settings provide global configurations that control the behavior of the approval workflow, such as setting priorities, escalation rules, and notifications.

- **Notification Preferences**- It defines how and when approvers are notified during the workflow.
- **Escalation Policies**- It specify the actions to take if an approver does not respond within a predefined time frame.
- **Audit Trails**- It enable detailed logging of all actions taken within the approval process to ensure accountability and provide reporting insights.

From business point of view approval workflow settings empower businesses to tailor workflows to meet specific requirements. It enhance traceability and accountability with comprehensive audit logs and minimize processing delays through robust notification and escalation mechanisms.

To create approval workflow settings refer to Defining Approval Workflow Group Settings

For more information Refer Approval Workflow Setting (Used for Searching)

Configuring Rules Engine for Polling Files on SFTP Server

The **Rules Engine for Polling Files on SFTP Server** allows automated and efficient handling of file uploads into the ORMB system. This ensures seamless file processing by applying predefined rules and criteria, improving accuracy and reducing manual intervention. From a business standpoint, configuring the rules engine for polling files on SFTP servers offers the following advantages:

- It automates file polling, reducing manual intervention and increasing operational efficiency.
- It provides robust error handling through email notifications or To-Do tasks, ensuring timely resolution of file-related issues.
- It allows organizations to adapt the rules engine to their specific workflows, enabling prioritization of critical data processing tasks.
- It supports large-scale operations with multi-threaded file processing capabilities, ensuring performance and reliability as data volume grows.

Rule Types

There are two predefined rule types used in SFTP poller batch execution:

File Upload Rule Type (FILE_RQ_TYPE) - The **File Upload Rule Type** defines the criteria for processing files that are uploaded to the ORMB staging area.

- **Input Parameters** - The following input parameters are used to evaluate and process files:
  - File Name (FILE_NAME) - It specifies the name of the file to be processed.
  - File Path (FILE_PATH) - It indicates the directory path where the file resides.
  - File Extension (FILE_EXTENSION) - It specifies the format of the file, such as `.csv`, `.xml`, or `.psv`.
  - Number Of Files (NUMBER_OF_FILES) - It identifies the total count of files present in the directory.
  - Directory Contains File (DIRECTORY_CONTAINS_FILE) - It provides the list of file names available in the directory.
- **Output Parameters** - The following output parameters are generated after evaluating the rule:
  - Job Name (JOB_NAME) - It specifies the batch job that should be executed upon successful validation.
  - Dependent Job Name (DEPENDENT_JOB_NAME) - It identifies prerequisite jobs that must execute before the current job.

Copyright © 2024, Oracle and/or its affiliates.

- Notification Parameters - It configures email notifications for stakeholders, including recipient details and content.

File Upload Rule Failure Type (FILE_UP_FAIL) - The **File Upload Rule Failure Type** is used to define rules for handling file polling failures. This rule type ensures that appropriate actions are taken for files that fail validation or processing. Notifications or corrective tasks can be triggered based on the rule configuration.

- **Behavior** - Email notifications can be sent to stakeholders when a file upload fails and To-Do tasks can be created to handle unresolved issues.
- **Parameters**:
  - **Input Parameters**- These include file metadata such as name, path, extension, and the number of files.
  - **Output Parameters**- It define failure-handling steps, such as triggering an alternate job or sending failure notifications.

Rules

Rules are defined based on conditions such as file type, directory path, or validation requirements. These criteria ensure that only eligible files are processed.

Example - A rule might specify that only `.xml` files larger than 1 MB should trigger the upload process.

Action Configuration - It specify what occurs when a rule evaluates to **True** or **False**

- **True Action** - It executes dependent jobs and marks the file processing as successful.
- **False Action** - It initiates retry mechanisms or alternative workflows and sends notifications for failures to stakeholders.

Setting the C1-FLUPLD Feature Configuration

You need to set the **C1-FLUPLD** feature configuration only when you want to decrypt the encrypted files on the SFTP server while uploading and transforming the files through the **C1-FTRAN** batch. The **C1-FRDALG** algorithm is used to decrypt the file using the keystore. Earlier, the following properties were derived from the `FW_spl.properties.keystore.truststore.include` file to encrypt or decrypt the files using the keystore:

- `com.oracle.ouaf.system.keystore.file`
- `com.oracle.ouaf.system.keystore.passwordFileName`
- `com.oracle.ouaf.system.keystore.type`
- `com.oracle.ouaf.system.keystore.padding`
- `com.oracle.ouaf.system.keystore.mode`
- `com.oracle.ouaf.system.keystore.alias`
- `ouaf.system.fileupload.pgp.pvt.key.filepath`

Now, the **C1-FLUPLD** feature configuration enables you to specify the required details to encrypt or decrypt the files using the keystore.

Procedure

To set the **C1-FLUPLD** feature configuration:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **F** and then click **Feature Configuration**.
   A sub-menu appears.
3. Click the **Search** option from the **Feature Configuration** sub-menu.

Copyright © 2024, Oracle and/or its affiliates.

The **Feature Configuration Query** screen appears.

4. In the **Feature Configuration Search** zone, enter **C1-FLUPLD** in the **Feature Name** field.

> **Note:** ORMB search engine supports wildcard search, where you can substitute the percentage (%) symbol as a stand in for any word or letter in a search criteria. You can use the '%' wildcard character in all input fields except the date and ID fields. The '%' wildcard character is suffixed automatically at the end of the partial search criteria. Therefore, you may or may not specify the wildcard character at the end of the partial search criteria. However, you have to prefix the wildcard character manually wherever required.

5. Click **Search**.
   A list of feature configurations that meet the search criteria appears in the search results.
6. In the **Search Results** section, click the link in the **Description** column corresponding to the feature configuration whose details you want to edit.

   The **Feature Configuration** screen appears. It contains the following option types:

Option Type	**Description**	**Mandatory (Yes or No)**
File Decryption Key Alias	Used to specify the alias (i.e. key defined in the keystore) using which you want to encrypt or decrypt the files. **Note:** By default, the value is set to **ouaf.system**. You can change the value as per the requirement.	Yes
File Decryption PGP Private Key File Path	Used to specify the directory or bucket from where you want to fetch the private key file for decrypting Pretty Good Privacy (PGP) files. If the application environment is an On Premise or Cloud Service environment, you can specify the directory name along with the following relative path: • **@SHARED_DIR** – Used when you want to fetch the private key file (for decrypting PGP files) from the shared directory on the server. However, if the application environment is a Cloud Service Premium Edition environment, you must specify the following information in the `file-storage:// <File Storage Extendable Lookup Value>/ <Object Storage Bucket>` format: • The file storage extendable lookup value from where the cloud storage compartment and connection details are retrieved while fetching the private key file (for decrypting PGP files) on OCI. • The bucket or container from where you want the system to fetch the private key file for decrypting PGP files. **Note:** By default, the value is set to **@SHARED_DIR/ PGP_KEYS/PGP_PRIVATE_KEY.asc**. You must change the value as per the requirement. The shared directory variable (i.e. **SHARED_DIR**) is case-sensitive.	Yes

Copyright © 2024, Oracle and/or its affiliates.

Option Type	Description	Mandatory (Yes or No)
File Decryption PGP Public Key Alias	Used to specify the alias (i.e. key in the public key file) using which you want to encrypt or decrypt the PGP files. **Note:** By default, the value is set to **ouaf.system.fileupload.publickey**. You can change the value as per the requirement.	Yes
File Decryption PGP Public Key File Path	Used to specify the directory or bucket from where you want to fetch the public key file for decrypting Pretty Good Privacy (PGP) files. If the application environment is an **On Premise** or **Cloud Service Environment**, you can specify the directory name along with the following relative path: • **@SHARED_DIR** – Used when you want to fetch the public key file (for decrypting PGP files) from the shared directory on the server. However, if the application environment is a Cloud Service Premium Edition environment, you must specify the following information in the `file-storage:// <File Storage Extendable Lookup Value>/ <Object Storage Bucket>` format: • The file storage extendable lookup value from where the cloud storage compartment and connection details are retrieved while fetching the public key file (for decrypting PGP files) on OCI. • The bucket or container from where you want the system to fetch the public key file for decrypting PGP files. **Note:** By default, the value is set to **@SHARED_DIR/ PGP_KEYS/PUBLIC_KEY.asc**. You must change the value as per the requirement. The shared directory variable (i.e. **SHARED_DIR**) is case-sensitive.	Yes
Key Store File Path	Used to specify the directory or bucket from where you want to fetch the keystore file. If the application environment is an On Premise or Cloud Service environment, you can specify the directory name along with the following relative path: • **@SHARED_DIR** - Used when you want to fetch the keystore file from the shared directory on the server. For example, you can set the value to `@SHARED_DIR/.ouaf_keystore`. However, if the application environment is a Cloud Service Premium Edition environment, you must specify	Yes

Copyright © 2024, Oracle and/or its affiliates.

Option Type	Description	Mandatory (Yes or No)
	the following information in the `file-storage://<File Storage Extendable Lookup Value>/<Object Storage Bucket>` format: • The file storage extendable lookup value from where the cloud storage compartment and connection details are retrieved while fetching the keystore file on OCI. • The bucket or container from where you want the system to fetch the keystore file. **Note:** By default, the value is set to **@SHARED_DIR/ks/.ouaf_keystore**. You must change the value as per the requirement. The shared directory variable (i.e. **SHARED_DIR**) is case-sensitive.	
Key Store Mode	Used to specify the mode using which you want to encrypt or decrypt the files. **Note:** By default, the value is set to **CBC** (i.e. Cipher-block Chaining).	Yes
Key Store Padding	Used to specify the padding using which you want to encrypt or decrypt the files. **Note:** By default, the value is set to **PKCS5Padding**.	Yes
Key Store Password File Path	Used to specify the directory or bucket from where you want to fetch the keystore password file. If the application environment is an On Premise or Cloud Service environment, you can specify the directory name along with the following relative path: • **@SHARED_DIR** - Used when you want to fetch the keystore password file from the shared directory on the server. For example, you can set the value to `@SHARED_DIR/.ouaf_storepass`. However, if the application environment is a Cloud Service Premium Edition environment, you must specify the following information in the `file-storage://<File Storage Extendable Lookup Value>/<Object Storage Bucket>` format: • The file storage extendable lookup value from where the cloud storage compartment and connection details are retrieved while fetching the keystore password file on OCI. • • The bucket or container from where you want the system to fetch the keystore password file.	Yes

Copyright © 2024, Oracle and/or its affiliates.

Option Type	Description	Mandatory (Yes or No)
	Note: By default, the value is set to **@SHARED_DIR/ks/.ouaf_storepass**. You must change the value as per the requirement. The shared directory variable (i.e. **SHARED_DIR**) is case-sensitive.	
Key Store Type	Used to indicate the type of keystore file. **Note:** By default, the value is set to **JCEKS**.	Yes
Private Key File Path	Used to specify the directory or bucket from where you want to fetch the private key file. If the application environment is an On Premise or Cloud Service environment, you can specify the directory name along with the following relative path: • **@SHARED_DIR** - Used when you want to fetch the private key file from the shared directory on the server. For example, you can set the value to `@SHARED_DIR/.ouaf_storepass`. However, if the application environment is a Cloud Service Premium Edition environment, you must specify the following information in the `file-storage:// <File Storage Extendable Lookup Value>/ <Object Storage Bucket>` format: • The file storage extendable lookup value from where the cloud storage compartment and connection details are retrieved while fetching the private key file on OCI. • The bucket or container from where you want the system to fetch the private key file. **Note:** By default, the value is set to **@SHARED_DIR/ks/.ouaf_storepass**. You must change the value as per the requirement. The shared directory variable (i.e. **SHARED_DIR**) is case-sensitive.	Yes

7. Enter the values for the required option types in the **Feature Configuration** screen.
8. Click the **Save** button in the **Page Title** area.
   The changes made to the feature configuration are saved.

Setting the File Upload Interface Master Configuration

Procedure

Copyright © 2024, Oracle and/or its affiliates.

To set the file upload interface master configuration:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **M** and then click **Master Configuration**.
   The **Master Configuration** screen appears.
3. In the **Master Configuration** zone, click the **Edit** (✎) icon in the **Edit** column corresponding to **File Upload Interface Configuration**.

   The **Master Configuration** screen appears. It contains the following fields:

Field Name	**Field Description**	**Mandatory (Yes or No)**
Master Configuration	Displays the master configuration. **Note:** This is a read-only field.	Not applicable
Validate Checksum	Used to indicate whether you want the system to check the integrity of the file before staging its contents in ORMB. If you select this option, the system will perform the checksum validation using the checksum file available on the SFTP server when the file validation algorithm **C1-FRHFVA** attached to the respective file request type is executed. However, if you do not select this option, the system will not perform the checksum validation for the files uploaded through the **File Upload Interface** feature. **Note:** The checksum file is available for the file on the SFTP server. The name of the checksum file should be identical to the name of the file that you want to upload. You must use the following naming convention for the checksum file - <FILE_NAME>.checksum. The **C1-FRHFVA** algorithm should be attached to the **File Validation Algorithm** field in the respective file request type. The checksum file is created using the md5sum command. During the checksum validation, the system checks whether the checksum value generated for the file in ORMB matches the checksum value specified in the checksum file. If so, the validation is successful. Otherwise, an error occurs indicating that the checksum value is incorrect. If you update a file after creating the checksum value for the file, you need to update the checksum value in the checksum file before uploading the file in the staging area of ORMB.	No
Validate Duplicate File Name	Used to indicate that a validation check for duplicate file-name is performed before the file is uploaded in the ORMB system. **Note:** If this field is unchecked, file with same name can be uploaded multiple times.	No
Audit Log Required	Used to indicate whether to log corresponding status changes of an individual file request after processing.	No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: Logging is done in CI_FILE_REQUEST_DTL_MSG table with LOG_ENTRY_TYPE_FLG value as **F1ST** CI_FILE_REQUEST_DTL_MSG table is also used for error logging with LOG_ENTRY_TYPE_FLG value as **F1EX**.	
File Name Size	Used to specify the maximum allowed size of a file names.	Yes (Conditional) **Note:** This field is required when you want to set the maximum length for a file name.
Max File Records	Used to specify the maximum number of records a file can contain during the upload process.	Yes (Conditional) **Note:** This field is required when you want to set the total number of records per file.
Max File Processing Size (in MB)	Used to specify the maximum size (in megabytes) of the file that can be processed. **Note:** For plain files, the file size should not exceed 2 GB (2000 MB) and for an encrypted file, the file size should not exceed 1 GB (1000 MB)	Yes (Conditional) **Note:** This field is required when you want to set the maximum size of a file to process.
Archive File	Used to indicate whether to archive the file or not once the corresponding file is processed. **Note:** You need to specify the location that refers to the path mentioned in the **Archive File Location** or **Archive Error File Location** field.	No
Archive File Location	Used to specify the file path that is needed to store the file(s) which are marked for archival. **Note:** **CFS Environment File Path**: The bucket path for storing files in the CFS environment is specified as `file-storage://ORMB_SHARED/DEV_IMPORT`. **On-Premise File Path**: The shared directory for file uploads in an on-premises setup is defined as `@SHARED_DIR/fileupload`.	Yes (Conditional) **Note:** This field is required when the **Archive File** option is selected.
Archive Error File Location	Used to specify the file path that is needed to store the file(s) which have been failed during the upload process.	Yes (Conditional)

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: **CFS Environment File Path**: The bucket path for storing files in the CFS environment is specified as `file-storage://ORMB_SHARED/DEV_IMPORT`. **On-Premise File Path**: The shared directory for file uploads in an on-premises setup is defined as `@SHARED_DIR/fileupload`.	**Note:** This field is required when the **Archive File** option is selected.
File Decryption Required	Used to indicate whether decryption is necessary for the uploaded file.	Yes (Conditional) **Note:** This field is required when you need to decrypt the file.
File Decryption Algorithm	Used to specify the algorithm that is used for decrypting the third party provided encrypted file before uploading in the ORMB application.	Yes (Conditional) **Note:** This field is required when the **File Decryption Flag** option is selected.
Cipher Type	Used to specify the type of cipher used for encryption or decryption. The valid values are: • Advanced Encryption Standard • Data Encryption Standard • Pretty Good Privacy • RSA with AES **Note:** It is used to get the encrypted keystore password that is required to read **ouaf_keystore** file which holds the encrypted file key.	Yes (Conditional) **Note:** This field is required when the encryption or decryption is required.
Private Key	Used to specify the key required for encryption or decryption.	Yes (Conditional)
	Note: It is updated in **ouaf_keystore** file, as it does not reside in the database. This field appears only when you select the **Advanced Encryption Standards**, **Data Encryption Standard**, or the **RSA with AES** option from the **Cipher Type** list.	**Note:** This field is required when the encryption or decryption is required.
Pass Phrase	Used to specify the pass phrase required for decryption.	Yes (Conditional)
	Note: This field appears only when you select the **Pretty Good Privacy** option from the **Cipher Type** list.	**Note:** This field is required when the **File Decryption Flag** option is selected.

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
File Encryption Required	Used to indicate whether encryption is necessary for the processed file.	Yes (Conditional) **Note:** This field is required when you need to encrypt the file.
File Encryption Algorithm	Used to specify the algorithm that is used for encrypt the third party provided decrypted file before uploading in the ORMB application.	Yes (Conditional) **Note:** This field is required when the **File Encryption Flag** option is selected.
Encryption Pass Phrase	Used to specify the pass phrase required for encryption. **Note:** This field appears only when you select the **Pretty Good Privacy** option from the **Cipher Type** list.	Yes (Conditional) **Note:** This field is required when the **File Encryption Flag** option is selected.
Upload Private Key	Used to provide an option to upload a private key file required for decryption or encryption. **Note:** This field appears only when you select the **Pretty Good Privacy** option from the **Cipher Type** list.	Yes (Conditional) **Note:** This field is required when the encryption or decryption is required.

Note: You can also set the file upload interface master configuration by clicking the **Edit** button in the **Master Configuration Details** zone.

4. Enter the required details in the **Main** section.
5. Click **Save**.
   The file upload interface master configuration is set.

Setting the Email Sender Master Configuration

Procedure

To set the email sender master configuration:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **M** and then click **Master Configuration**.
   The **Master Configuration** screen appears.
3. In the **Master Configuration** zone, click the **Edit** (✎) icon in the **Edit** column corresponding to the **Email Sender Configuration**.

   The **Master Configuration** screen appears. It contains the following fields:

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
Master Configuration	Displays the master configuration. **Note:** This is a read-only field.	Not applicable
SMTP Host Name	Used to specify the SMTP application to send and receive emails. **Note:** By default **internal-mail-router.oraclecorp.com** host-name is present in the system.	Yes
SMTP Port	Used to specify the communication endpoint to transfer email data over SMTP through a network, from one server to another. The valid value are: • 465 • 587 • 25 **Note:** The default SMTP port is set to **25**.	Yes
Username	Used to specify the username for the user.	No
Password	Used to specify the password for the user.	No
Secure Sockets Layer	Used to indicate whether you want to transfer the email data securely across a network.	No
Test	Used to configure the sender's and receiver's email ID.	No
From Email Address	Used to specify the user's email ID. **Note:** This field appears only when you select the check box corresponding to the Test field.	No
To Recipient Email ID	Used to specify the receiver's email ID. **Note:** This field appears only when you select the check box corresponding to the Test field.	No

Note: You can also edit the master configuration details on the **Master Configuration** screen by clicking the **Edit** button in the **Master Configuration Details** zone.

4. Enter the required details in the **Main** section.
5. Click **Save**.
   The changes made to the master configuration are set.

Setting the File Management System Configuration

The **File Management System** master configuration enables you to set certain options which are used while uploading or downloading files from the SFTP server.

Procedure

Copyright © 2024, Oracle and/or its affiliates.

To set the **File Management System** master configuration:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **M** and then click **Master Configuration**.
   The **Master Configuration Query** screen appears.
3. In the **Master Configuration Search** zone, click the **File Management System Configuration** link in the **Master Configuration** column.
   The **Master Configuration** screen appears.
4. Click the **Edit** button in the **Master Configuration** zone.

   The **Master Configuration** screen appears. It contains the following section:

   - **Main** - Used to specify the basic details for the file management system master configuration. It contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Upload File Directory	Used to specify the directory path of the SFTP server. The system will then accordingly lists the files and subdirectories from the specified directory in the **Search File or Directory** zone when the file path criteria is not specified. **Note:** By default, the value is set to **@SHARED_DIR**. You can change the value, if required. However, the directory should exists inside **@SHARED_DIR**. For example, @SHARED_DIR/autopay. **@INSTALL_DIR** is deprecated and no longer supported.	Yes
Upload Maximum File Size (MB)	Displays the maximum file size that you can upload. **Note:** By default, the maximum file size is set to 10 MB.	Not applicable
Upload Maximum File Name Length	Displays the maximum length of a filename that is supported while uploading the file. **Note:** By default, the maximum file name length is set to 60 characters.	Not applicable

5. Modify the required details in the **Main** section.
6. Click **Save**.
   The changes made to the master configuration are saved.

File Management System

Oracle Revenue Management and Billing enables you to upload and download files from the shared directory on the SFTP server. The **File Management System** screen allows you to do the following:

- Search a file or a directory on the shared directory of the SFTP server.
- View the files on the shared directory which are yet to be uploaded in the ORMB staging tables.
- View the files whose upload process on the SFTP server is in progress.

Copyright © 2024, Oracle and/or its affiliates.

- View the files which are already uploaded in the ORMB staging tables and are currently awaiting approval for processing.
- View the files which are already uploaded in the ORMB staging tables but yet to be processed.
- Upload a file from the local or network machine to the shared directory.
- Download a file from the shared directory to the local machine.

This screen contains the following zone:

- Search File or Directory on page 953

Related Topics

For more information on...	See...
How to search for a file or a directory	Searching for a File or a Directory on page 957
How to view the files of a directory	Viewing the Files of a Directory on page 957
How to download a file from the SFTP server	Downloading a File from the SFTP Server on page 958
How to upload a file to the SFTP server	Uploading a File to the SFTP Server on page 958

Search File or Directory

The **Search File or Directory** zone allows you to search for a file or a directory on the SFTP server using various search criteria. It contains the following two sections:

- **Search Criteria** - The **Search Criteria** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
File Name	Used to search for files with a particular file name.	No
File Status	Used to search for files with a particular status. The valid values are: • **Ready to Upload** - Used when you want to view the files on the SFTP server which are yet to be uploaded in the ORMB staging tables. • **Copy in Progress** - Used when you want to view the files whose upload process on the SFTP server is in progress. • **Approval Pending** - Used when you want to view the files which are already uploaded in the ORMB staging tables and are currently awaiting approval for processing. • **Pending to Process** - Used when you want to view the files which are already uploaded in the ORMB staging tables but yet to be processed. **Note:** By default, the **Ready to Upload** option is selected.	Yes
File Request Type	Used to search for files which are already uploaded in the ORMB staging tables using a particular file request type.	No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: The list does not include all the file request types which are defined in the system. Instead, the system considers the files which are either in **Pending** or **Approval Pending** status in the staging tables and accordingly lists the respective file request types.	
File Path	Used to search for files which are uploaded on the SFTP server in a particular directory. The valid value is: • @SHARED_DIR **Note:** You must specify the path in uppercase. Otherwise, erroneous results might occur. If you do not specify the file path, the system retrieves the files from the directory specified in the **Upload File Directory** field of the **C1-FileManagementConfig** master configuration. For more information, see Setting the File Management System Configuration on page 951.	No
File Upload From Date	Used to search for files which are uploaded from a particular date onwards. **Note:** You can either manually specify the date or select it using the **Date Picker** ( ) icon corresponding to the field.	No
File Upload To Date	Used to search for files which are uploaded till a particular date. **Note:** You can either manually specify the date or select it using the **Date Picker** ( ) icon corresponding to the field.	No
File Type	Used to indicate whether you want to view the files or directories on the SFTP server. The valid values are: • File • Directory	No

Note: You must specify at least one search criterion while searching for a file.

- **Search Results** - On clicking the **Search** button, the search results appear based on the specified search criteria. The **Search Results** section contains the following columns:

Column Name	Column Description
File Name	Displays the file or directory name.

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
	Note: It has a link. On clicking the link, the respective screen appears: • **File Upload Dashboard** - This screen appears when you click the link in the **File Name** column corresponding to a file which is in the **Approval Pending** or **Pending To Process** status. • **File Management System** - This screen appears with a list of files (within the directory) when you click the link in the **File Name** column corresponding to a directory. If you click the link in the **File Name** column corresponding to a file which is in the **Ready To Upload** status, the system redirects you to the **File Management System** screen itself.
File Path	Indicates the path where the file is uploaded on the SFTP server.
	Note: This column appears when the **Ready To Upload** option is selected from the **File Status** list. In case of directory, the column value is set to **Directory**.
File Size (KB)	Indicates the size of the file in Kilobyte (KB).
	Note: This column appears when the **Ready To Upload** option is selected from the **File Status** list. In case of directory, no data appears in this column.
File Request Type	Indicates the file request type using which the file is uploaded in the ORMB staging tables.
	Note: This column appears when the **Approval Pending** or **Pending To Process** option is selected from the **File Status** list. It has a link. On clicking the link, the **File Request Type** screen appears where you can view the details of the respective file request type.
File Status	Indicates the status of the file. The valid values are: • Approval Pending • Pending to Process
	Note: This column appears when the **Approval Pending** or **Pending To Process** option is selected from the **File Status** list.
File Upload Approval Required	Indicates whether approval is required before processing a file. The valid values are: • YES • NO

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
	Note: This column appears when the **Approval Pending** or **Pending To Process** option is selected from the **File Status** list.
Approval Transaction ID	Displays the approval transaction ID.
	Note: It has a link. On clicking the link, the **Approve** screen appears where you can view the details of the respective approval transaction. This column appears when the **Approval Pending** or **Pending To Process** option is selected from the **File Status** list. However, note that no data appears in this column when the file is in the **Pending To Process** status.
Upload Date Time	Displays the date and time when the file is uploaded in the system.
	Note: In case of directory, no data appears in this column.
Download File	On clicking the **Download** link, you can download the file.
	Note: This column appears when the **Ready To Upload** option is selected from the **File Status** list. In case of directory, no data appears in this column.
Action	Indicates the type of action, such as **Move** or **Upload** that you can perform for the file or directory, respectively.
	Note: This column appears when the **Ready To Upload** option is selected from the **File Status** list. It has a link. On clicking the link, the **Action** screen appears from where you can upload or move a file. At present, the **Move** action is not operational as the @INSTALL_DIR is deprecated and no longer supported.

Note: Pagination is used to display limited number of records in the **Search Results** section. You can use the navigation links, such as **Previous** and **Next** to navigate between pages.

Related Topics

For more information on...	See...
How to search for a file or a directory	Searching for a File or a Directory on page 957
How to view the files of a directory	Viewing the Files of a Directory on page 957
How to download a file from the SFTP server	Downloading a File from the SFTP Server on page 958
How to upload a file to the SFTP server	Uploading a File to the SFTP Server on page 958

Copyright © 2024, Oracle and/or its affiliates.

Searching for a File or a Directory

Procedure

To search for a file or a directory:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **F** and then click **File Management System**.
   The **File Management System** screen appears.
3. Enter the search criteria in the **Search File or Directory** zone.

   Note: ORMB search engine supports wildcard search, where you can substitute the percentage (%) symbol as a stand in for any word or letter in a search criteria. You can use the '%' wildcard character in all input fields except the date and ID fields. The '%' wildcard character is suffixed automatically at the end of the partial search criteria. Therefore, you may or may not specify the wildcard character at the end of the partial search criteria. However, you have to prefix the wildcard character manually wherever required.
4. Click **Search**.
   A list of files or directories that meet the search criteria appears in the **Search Results** section.

Related Topics

For more information on...	See...
File Management System screen	File Management System on page 952
Search File or Directory zone	Search File or Directory on page 953

Viewing the Files of a Directory

Procedure

To view the files of a directory:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **F** and then click **File Management System**.
   The **File Management System** screen appears.
3. Select the **Directory** option from the **File Type** list.
4. Enter additional search criteria, if required, in the **Search File or Directory** zone.
5. Click **Search**.
   A list of directories that meet the search criteria appears in the search results.
6. In the **Search Results** section, click the link in the **File Name** column corresponding to a directory whose contents you want to view.
   The files and sub-directories of the respective directory are listed in the search results.

Related Topics

For more information on...	See...
File Management System screen	File Management System on page 952
Search File or Directory zone	Search File or Directory on page 953

Copyright © 2024, Oracle and/or its affiliates.

Downloading a File from the SFTP Server

Procedure

To download a file from the SFTP server:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **F** and then click **File Management System**.
   The **File Management System** screen appears.
3. Select the **File** option from the **File Type** list.
4. Enter additional search criteria, if required, in the **Search File or Directory** zone.
5. Click **Search**.
   A list of files that meet the search criteria appears in the search results.
6. In the **Search Results** section, click the link in the **Download File** column corresponding to the *f*ile that you want to download.
   The file is downloaded.

Note: The system allows you to download a file which does not exceed 10 MB.

Related Topics

For more information on...	See...
File Management System screen	File Management System on page 952
Search File or Directory zone	Search File or Directory on page 953

Uploading a File to the SFTP Server

Procedure

To upload a file to the SFTP server:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **F** and then click **File Management System**.
   The **File Management System** screen appears.
3. Select the **Directory** option from the **File Type** list.
4. Enter additional search criteria, if required, in the **Search File or Directory** zone.
5. Click **Search**.
   A list of directories that meet the search criteria appears in the search results.
6. In the **Search Results** section, click the **Upload** link in the **Action** column corresponding to the directory where you want to upload the file.

   The **Action** screen appears. It contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Upload File	On clicking the **Choose File** button, you can browse through the system and choose a file that you want to upload.	Yes

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: You cannot upload a file which is more than 10 MB. You can upload all types of files except archive files. A file name can be up to 60 characters long and can contain letters, numbers, and special characters, such as period (.), underscore (_), hyphen (-) and whitespace ( ).	
Replace Existing File	Used to indicate whether you want to replace any existing file in the system. You must select this option when you want to upload the same file once again.	No

7. Click the **Choose File** button corresponding to the **Upload File** field.
   The **Open** dialog box appears.
8. Browse to the location where the file that you want to upload is available in the system.
9. Click **Open**.
   The file name appears in the **Upload File** field.
10. Select the **Replace Existing File** option depending on whether you want to upload the same file once again.
11. Click **Upload**.
    A message appears indicating that the file is uploaded successfully on the SFTP server.
12. Click **OK**.

Related Topics

For more information on...	See...
File Management System screen	File Management System on page 952
Search File or Directory zone	Search File or Directory on page 953

File Request Type

The **File Request Type** screen allows you to configure the system for uploading the files of an individual screen, entity, or functionality (i.e., person creation, account creation, transaction creation, payment creation, or payment upload, etc.) in a format which is complaint with the formats supported by the application. It also enables you to transform the format of an uploaded file into a format supported by the application.

The **File Request Type** screen allows you to do the following:

- Define a file request type
- Edit a file request type
- Copy a file request type
- Delete a file request type
- View the details of a file request type

The **File Request Type** screen contains the following zones:

- File Request Type List on page 960
- File Request Type on page 960

Copyright © 2024, Oracle and/or its affiliates.

File Request Type List

The **File Request Type List** zone lists the file request types that are already defined in the system. It contains the following columns:

Column Name	Column Description
File Request Type	Displays the file request type.
Description	Displays the description of the file request type.
Edit	On clicking the **Edit** (✎) icon corresponding to a file request type, the **File Request Type** screen appears where you can edit the details of the file request type.
Delete	On clicking the **Delete** (🗑) icon corresponding to a file request type, you can delete the file request type.
	Note: You can delete a file request type only when a file upload request is not yet created using the file request type.
Duplicate	On clicking the **Duplicate** (⧉) icon corresponding to a file request type, the **File Request Type** screen appears where you can define a new file request type using an existing file request type.

You can filter the list using various search criteria (such as, **File Request Type** and **Description**) available in the **Filter** area. By default, the **Filter** area is hidden. You can view the **Filter** area by clicking the **Filters** (▽) icon in the upper right corner of this zone or by clicking the **Expand Filter** button available above the **File Request Type** list.

On clicking the **Broadcast** (📶) icon corresponding to a file request type, the **File Request Type** zone appears with the details of the respective file request type.

Note: Pagination is used to display limited number of records in this zone. You can use the navigation links, such as **Previous** and **Next** to navigate between pages.

Related Topics

For more information on...	See...
File Request Type zone	File Request Type on page 960
How to edit a file request type	Editing a File Request Type on page 981
How to copy a file request type	Copying a File Request Type on page 989
How to delete a file request type	Deleting a File Request Type on page 989

File Request Type

The **File Request Type** zone displays the details of the file request type. It contains the following sections:

- **Main** - The **Main** section displays basic information about the file request type. It contains the following fields:

Field Name	Field Description
File Request Type	Displays the file request type.
Description	Displays the description of the file request type.
Data Transformation Required	Indicates whether data transformation is required for the uploaded file.

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description
File Atomicity	Indicates whether all records in a file should be committed or rolled back in case of any error occurred while processing the uploaded file.
File Format	Displays the format in which the data is transformed for the uploaded file. The valid values are: • Comma Separated Values • Extensible Markup Language • Fixed Position • JavaScript Object Notation • Pipe Separated Values • Tilde Separated Values
File Extension	Displays the extension of the file format.
Upload and Process File Simultaneously	Indicates whether the upload and process of the file record takes place in a single batch run.
	Note: On selecting this option, the File Transform and Upload (C1-FTRAN) batch is executed where the upload and process of the file records takes place in a single batch run. Whereas, if this option is not selected, you need to execute the File Request Process (C1-FREQP) batch for processing the file records.
Root XML Tag	Displays the name of the root tag whose data is transformed in the xml format.
	Note: This field is displayed only when the **Data Transformation Required** option is selected and the **File Format** field is set to **Extensible Markup Language**.
File Header Required	Indicates whether the file should be uploaded/processed with header details.
	Note: If the file header required option is checked then the first record in the file will always be considered as the file header.
Header XML Tag	Displays the header tag element of the file. It is used to get the header element block from the corresponding XML file.
	Note: This field is displayed when the file header required and data transformation required option field is checked where the file format field option is selected as **Extensible Markup Language**.
File Footer Required	Indicates whether the file should be uploaded/ processed with footer details.
	Note: If file footer required option is checked, the last record in the file will always be considered as the file footer.
Footer XML Tag	Displays the footer tag element of the file. It is used to get the footer element block from the corresponding XML file.
	Note: This field is displayed when the file footer required and data transformation required option field is checked and the file format field option is selected as **Extensible Markup Language**.
Static Header	Indicates that the uploaded file contains a predefined, unchanging header row or tag, which does not need dynamic validation during processing.

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description
Static Footer	Indicates that the uploaded file contains a predefined, unchanging footer row or tag, typically used for summary or ending markers without requiring dynamic checks.
Service Log Required	Indicates whether we want to log the service execution details for every individual record in a file. **Note:** Same details will be used on **File Request dashboard** for navigating to its created or updated corresponding entity.
Skip Duplicates	Indicates if we want to have delta check i.e. avoid re-processing of same record in multiple batch executions or record level duplicate check i.e. if we want to skip the execution of those records for which we have multiple entries in a single file. Multiple entries in a file are identified using **Record Identifier** configured in **Transformation Details** section. **Note:** Multiple entries in a file are identified using the **Record Identifier** field, configured in the **Transformation Details** section.
File Validation Algorithm	Displays the file validation algorithm for header, footer, and checksum validation field. **Note:** It has a link. On clicking the link, the **Algorithm** screen appears where you can view the details of the respective file validation algorithm.
Data Transformation Algorithm	Displays the algorithm, used to transform the uploaded file to the required ORMB compliance service.
Maximum Retry for Error Record	Displays the maximum number of retry attempts that are allowed in case of a record failure.
Validate Record Payload	Indicates that only those records should be processed that have payloads for all those configured services of file request type.
Display Profile	Displays the display profile which is linked with the file request type. The valid values are: • Europe • North America • HIJRI Format • Other
File Upload Approval Required	Displays if this file request type corresponding file is uploaded using approval workflow.
Date Format	Displays the client supported date format. **Note:** This field becomes visible and mandatory only when **Other** option is selected as the **Display Profile**
Time Format	Displays the client supported time format.

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description
	Note: This field becomes visible and mandatory only when **Other** option is selected as the **Display Profile**
File Decryption Required	Indicates whether the uploaded file must be decrypted before being processed.
Duplicate Record Identifier Expression	Indicates the expression or logic used to identify duplicate records within the file.
Group Duplicate Records	Indicates whether duplicate records in the uploaded file should be grouped together for processing.
Duplicate Records Order By	Indicates the sorting order for duplicate records, determining how they are processed or grouped.
External System	Displays an external system for configuring a file request type.
	Note: This is an information based field.
Force File Archival	Indicates that the uploaded file must be archived (stored) after processing, regardless of other factors.
File Record Size Greater Than 32KB	Indicates if the record size is greater than 32KB for the file request type.
	Note: This field is ideally used for performance optimization. If the record size is greater than 32kb, then **REQUEST** field with **CLOB** data type will be used to capture file record payload, else record payload will be staged in **BO_DATA_AREA** field with **VARCHAR2(32000)** data type.
Error To Do Type	Indicates the type of task **To Do** to be created for handling errors encountered during file processing.
	Note: The **Search** (🔍) icon appears corresponding to the **Error To Do Type** field. On clicking the search icon, the **To do Type Details** search window appears.
Detailed Description	Displays additional information about the file request type.

- **Services** - The **Services** section lists the service defined for the file request type. It contains the following fields:

Field Name	Field Description
Sequence	Displays the sequence of service invocation and execution.
Service Type	Indicates the type of service. The valid values are: • Business Object • Business Service • Service Script
Service Name	Displays the name of the service to be triggered and executed for each processing record.
FK Reference	Displays the foreign key reference code from where the added or updated details can be viewed. It has a link. On clicking the link, the **Foreign Key Reference** screen appears where you can view the details of the respective foreign key reference.

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description
Pre-Processing Algorithm	Displays the algorithm required for pre-processing tasks before running the service. It has a link. On clicking the link, the **Algorithm** screen appears where you can view the details of the respective pre-processing algorithm.
Post-Processing Algorithm	Displays the algorithm to be executed after a record has been successfully processed. It has a link. On clicking the link, the **Algorithm** screen appears where you can view the details of the respective post-processing algorithm.
Operation	Indicates the operation to be performed for the individual service (**Business Object**). The valid values are: • Add • Update • Replace • Delete
Dependent Service Name	Displays the payload hierarchical relationship and manage dependent services, indicating a parent-child association.
Defer Completion	Indicates a flag that delays the final status update for a successfully processed request until the life-cycle of the corresponding **Business Object** is complete.
	Note: An API is provided to update the final status of the respective request.

- **Messages** - The **Messages** section lists the message defined for the file request type. It contains the following columns:

Column Name	Column Description
Message Category	Displays the message category number.
	Note: The **Search** (🔍) icon appears corresponding to the **Message Category** field. On clicking the search icon, the **Message Category Search** window appears.
Message Number	Displays the number to handle the **Retry** or **Sent for Approval** status.
	Note: The **Search** (🔍) icon appears corresponding to the **Message Number** field. On clicking the search icon, the **Fetch File Request Type Message Number** window appears.
Message Text	Indicates description of the message.
	Note: This is read-only field. As the **Message Number** field is interdependent on **Message Category**, if you search only for **Message Number**, no text will appear in the **Message Text** field. You have to first enter the **Message Category**, and then choose the **Message Number**.
File Record Status	Displays the status for every file record. The valid values are: • Retry • Sent for Approval

- **Data Transformation** - The **Data Transformation** section lists the data transformation details defined for the file request type. It is divided into below sub-sections:

Copyright © 2024, Oracle and/or its affiliates.

- **Header Transformation** - The **Header Transformation** sub-section contains the following columns:

Column Name	Column Description
Sequence	Displays the sequence of the header values in a file.
Field Name	Displays a unique name for the header field that will map to the corresponding file sequence header value.
Required	Indicates the header level validations are necessary for every record while uploading data using the **File Transform and Upload (C1-FTRAN)** batch.
Edit Input	Displays whether the header value can be modified during processing, based on the default value configurations.
Default Value	Displays the default values that will be applied when generating the record payload.
Date/Time Format	Displays the format of date or time values in the input file according to client-defined formats.

- **Footer Transformation** - The **Footer Transformation** sub-section contains the following columns:

Column Name	Column Description
Sequence	Displays the sequence of the footer values in a file.
Field Name	Displays a unique name for the footer field that will map to the corresponding file sequence footer value.
Required	Indicates the footer level validations are necessary for every record while uploading data using the **File Transform and Upload (C1-FTRAN)** batch.
Edit Input	Displays whether the footer value can be modified during processing, based on the default value configurations.
Default Value	Displays the default values that will be applied when generating the record payload.
Date/Time Format	Displays the format of date or time values in the input file according to client-defined formats.

- **Field Transformation** - This sub-section contains the following columns:

Column Name	Column Description
Sequence	Displays the sequence of the field values in a file.
Field Name	Displays a unique name for the field that will map to the corresponding file sequence field value.
Source Field Path	Displays the XPath expression that references elements in XML or JSON files from the legacy system.
Map Field XPath	Displays an XPath expression syntax to map elements in business objects during transformation. **Note:** The **Search** (🔍) icon appears corresponding to the **Map Field XPath** field. On clicking the search icon, the **Field XPath Search** window appears.
Required	Indicates that field level validations are necessary for every record while uploading the data, using the **File Transform and Upload (C1-FTRAN)** batch.

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
Record Identifier	Indicates whether the records should be uniquely displayed in the file upload dashboard or not.
Skip Validation	Indicates whether validation should be skipped as per metadata configuration.
Edit Input	Displays whether the field value can be edited during processing.
Default Value	Displays the default values for the fields while the record payload is getting generated.
Date/Time Format	Displays the format of date or time values in the input file according to client-defined formats.

Note: By default, the **File Request Type** zone does not appear in the **File Request Type** screen. It appears when you click the **Broadcast** ( ) icon corresponding to a file request type in the **File Request Type List** zone.

Related Topics

For more information on...	See...
File Request Type screen	File Request Type on page 959
File Request Type List zone	File Request Type List on page 960

Defining a File Request Type

Prerequisites

To define a file request type, you should have:

- File request type business objects defined in the application

Procedure

To define a file request type:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **F** and then click **File Request Type**.
   A sub-menu appears.
3. Click the **Add** option from the **File Request Type** sub-menu.
   The **File Request Type** screen appears. It contains the following sections:
   - **Main** - Used to specify basic configuration details about the file request type.
   - **Services** - Used to add services in the file request type.
   - **Messages** - Used to specify the messages associated with the file request type.
   - **Data Transformation** - Used to specify the header, footer, and field transformation details in the file request type.

   Note: This section appears only when you select the **Data Transformation Required** field from the **Main** section.

   The **Main** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
File Request Type	Used to specify the file request type.	Yes
Description	Used to specify the description for the file request type.	No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Tip: You need to specify the purpose of the file request type.	
Data Transformation Required	Used to indicate that the type of file or schema which you want to upload is incompatible and needs transformation as per the type of file or schema supported by the application. **Note:** The application supports file type or schema which is in the Extensible Markup Language (XML) format. Therefore, if you are uploading a file or schema in the Extensible Markup Language (XML), you need not select the **Data Transformation Required** field. However, if you are uploading a file or schema which is in a format other than the Extensible Markup Language (XML), you need to select the **Data Transformation Required** field.	Yes (Conditional) **Note:** This field is required while transforming the data received from the external source (i.e. from the customer) as per the format provided in the application.
File Atomicity	Used to indicate whether the system should stop the file upload process and roll back the entire data if any error occurs in any single record of the uploaded file.	No
File Format	Used to indicate the type of file that you want to upload using the file request type. The valid values are: • Comma Separated Values • Extensible Markup Language • Fixed Position • JavaScript Object Notation • Pipe Separated Values • Tilde Separated Values **Note:** By default, the **Extensible Markup Language** option is selected and the field is disabled. It is enabled only when you select the **Data Transformation Required** field.	Yes
File Extension	Used to specify the extension of the file type selected from the **File Format** list. **Note:** This field is enabled only when you select the **Data Transformation Required** field.	Yes
Upload and Process File Simultaneously	Used to indicate that you want to upload and process the file simultaneously in a single batch run.	No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: On selecting this field, when the File Transform and Upload (C1-FTRAN) batch is run, the system will upload the files and create a staging table for the records available in the uploaded file. These records will then be assigned to the respective tables of the entity in the database. However, if you do not select this field, the File Transform and Upload (C1-FTRAN) batch will only upload the file and create a staging table for the records available in the uploaded file. To assign the data from the staging table to the respective tables of the entity in the database you need to run the File Request Process (C1-FREQP) batch. This field is selected automatically when the **File Atomicity** field is selected.	
Root XML Tag	Used to specify the root tag element in the file which you want to upload using the file request type. **Note:** This field is displayed only when the **Data Transformation Required** filed is selected and the **File Format** field is set to **Extensible Markup Language**.	No
File Header Required	Used to indicate whether the file that you want to upload should have header details or not. **Note:** If you select this field, the first row data in the uploaded file will be considered as the file header string. If the **Data Transformation Required** filed is selected, you need to configure the header details in the **Header Transformation** sub-section of the **Data Transformation** section. The **Static Header** field is automatically deselected if you select the **File Header Required** field.	No
Header XML Tag	Used to specify the header tag element in the file whose data you want to use as the header details. **Note:** This field is displayed only when the **Data Transformation Required**, **File Header Required** fields are selected and the **File Format** field is set to **Extensible Markup Language**.	Yes (Conditional) **Note:** This field is required while uploading an Extensible Markup Language (XML) file for which data transformation is required.
File Footer Required	Used to indicate whether the file that you want to upload should have footer details or not.	No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: If you select this field, the last row data in the uploaded file will be considered as the file footer string. If the **Data Transformation Required** filed is selected, you need to configure the footer details in the **Footer Transformation** sub-section of the **Data Transformation** section. The **Static Footer** field is automatically deselected if you select the **File Footer Required** field.	
Footer XML Tag	Used to specify the footer tag element in the file whose data you want to use as the footer details. **Note:** This field is displayed only when the **Data Transformation Required**, **File Footer Required** fields are selected and the **File Format** field is set to **Extensible Markup Language**.	Yes (Conditional) **Note:** This field is required while uploading an Extensible Markup Language (XML) file for which data transformation and footer details is required.
Static Header	Used to indicate that the uploaded file contains a predefined, unchanging header row which does not need dynamic validation during processing. **Note:** If the **Data Transformation Required** filed is selected, you need to configure the static header details in the **Header Transformation** sub-section of the **Data Transformation** section. The **File Header Required** field is automatically deselected if you select the **Static Header** field.	No
Static Footer	Used to indicate that the uploaded file contains a predefined, unchanging footer row typically used for summary or ending markers without requiring dynamic validation during processing. **Note:** If the **Data Transformation Required** filed is selected, you need to configure the static footer details in the **Footer Transformation** sub-section of the **Data Transformation** section. The **File Footer Required** field is automatically deselected if you select the **Static Footer** field.	No
Service Log Required	Used to indicate whether you want to log details about the creation and updates made in every record of the entity	No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	corresponding to the services configured for the file request type. **Note:** The service log details are used on the **File Request dashboard** screen for navigating to the changes made in the corresponding entity.	
Skip Duplicates	Used to indicate whether you want to skip the execution of duplicate records in the uploaded file. **Note:** If you select this option, all the duplicate records will be marked as **IGN** and none of the duplicate record will be processed by the batch. For Example: If you have two duplicate records in the uploaded file and the **Skip Duplicate** field is selected, then, the system will skip both the records while processing the file. Duplicate entries are identified when you select the **Record Identifier** field corresponding to the **Field Name** configured in the **Field Transformation** sub-section of the **Data Transformation** section.	No
File Validation Algorithm	Used to specify the algorithm which you want to use for validating the data in the uploaded file. The valid values are: • **File Request Header Footer Validation** (C1-FRHFVA) - Used to validate the header and footer provided in the file. If the **File Header Required** or **Static Header** field is selected, the algorithm checks whether the header line in the file is marked with the header marker **H**. If the **File Footer Required** or **Static Footer** field is selected, the algorithm checks whether the footer line in the file is marked with the footer marker **T**, followed by the count of records in the file. **Note:** If none of the above mentioned conditions are met, the system will stop processing the file. • **Sample File Validation** (C1-FRHVA) - Used to check whether the associated algorithm is invoked and logs the fields available in the header and footer while processing the file.	Yes (Conditional) **Note:** This field is required for validating the header and footer information while processing the data provided in the file using the file request type.

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: The **Search** (🔍) icon appears corresponding to the **File Validation Algorithm** field. On clicking the search icon, the **Algorithm Search** search window appears where you can search for the respective algorithm. The above algorithms are shipped with the product. The implementation team can create custom algorithm type for this algorithm entity based on the customer requirements.	
Data Transformation Algorithm	Used to indicate the algorithm using which the data from the uploaded file is transformed into data which is compliant with the schema defined for the business service in the application. **Note:** The **Search** (🔍) icon appears corresponding to the **Data Transformation Algorithm** field. On clicking the search icon, the **Algorithm Search** search window appears where you can search for the respective algorithm. This field appears only when you select the **Data Transformation Required** field.	Yes (Conditional) **Note:** This field is required while data transformation.
Maximum Retry for Error Record	Used to specify the maximum number of retry attempts that are allowed for records which are in the error status while processing the uploaded file.	No
Validate Record Payload	Used to indicate that you want to validate and process only those records that have payloads for other services that are configured using the file request type.	No
Display Profile	Used to indicate how the date should be displayed when the entity is created. The valid values are: • Europe • North America • HIJRI Format • Other	No
File Upload Approval Required	Used to indicate whether approval is required to upload and process the file using the file request type.	No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: If you select this option you need to ensure that the approval workflow for file upload interface is configured in the system. When the File Transform and Upload (C1-FTRAN) batch is run, the system will create an Approval Transaction ID and mark the file in the **Pending for Approval** status. The approver can then search and approve the file from the **Approval Transaction** screen using the Approval Transaction ID. Once the approver approves the file from the **Approval Transaction** screen, the File Transform and Upload (C1-FTRAN) batch is triggered automatically for processing the file.	
Date Format	Used to specify the format in which you want to display the date. **Note:** This field appears only when the **Other** option is selected from the **Display Profile** list.	No
Time Format	Used to specify the format in which you want to display the time. **Note:** This field appears only when the **Other** option is selected from the **Display Profile** list.	No
File Decryption Required	Used to indicate whether decryption is required for the file before processing. **Note:** If you select this field, the system will check for the encryption details configured for the file upload interface in the **Master Configuration** screen.	No
Duplicate Record Identifier Expression	Used to specify the field for which you want to identify duplicate records in the uploaded file. **Note:** If you want to specify multiple fields, you can do so by adding <space>&&<space> as the separator in between the fields. If you want to specify a combination of fields, you can do so by adding <space>||<space> as the separator in between the fields.	No
Group Duplicate Records	Used to indicate whether duplicate records in the uploaded files should be processed in a same thread or not.	No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: If you select this option, all the duplicates will be processed as a single work unit. Hence, processed record count will be less than the total record count.	
Duplicate Records Order By	Used to indicate the order in which you want to process the duplicate records.	No
External System	Used to specify an external system for configuring a file request type.	No
Force File Archival	Used to indicate whether the file should be archived and moved to the location mentioned in the **Master Configuration** screen after processing.	No
Record Size Greater Than 32KB	Used to indicate whether each record in the file which you want to upload using the file request type is greater than 32 KB or not. **Note:** Record size greater than 32 KB will be stored in the **CLOB** data type. Whereas, record size less than 32 KB, will be stored in **VARCHAR2** data type in the database.	No
Error To Do Type	Used to indicate the To Do entry of the specified To Do Type must be created in case of any error encountered while processing the file using the file request type. **Note:** The **Search** (🔍) icon appears corresponding to the **Error To Do Type** field. On clicking the search icon, the **To Do Type Details** window appears where you can search for the respective To Do Type.	No
Detailed Description	Used to specify additional information about the file request type.	No

Tip: Alternatively, you can access this screen by clicking the **Add** button in the **Page Title** area of **File Request Type** screen.

4. Enter the required details in the **Main** section.
5. Add, edit, or delete services from the file request type, if required.
6. Add, edit, or delete messages from the file request type, if required.
7. Add, edit, or delete the data transformation details from file request type, if required.
8. Click **Save**.
   The **File Request Type** is defined.

Related Topics

For more information on...	See...
File Request Type screen	File Request Type on page 959
How to add services in a file request type	Adding Services in a File Request Type on page 974
How to add messages in a file request type	Adding Messages in a File Request Type on page 975
How to add header data transformation details in a file request type	Adding Header Data Transformation Details in a File Request Type on page 976

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
How to add footer data transformation details in a file request type	Adding Footer Data Transformation Details in a File Request Type on page 978
How to add field data transformation details in a file request type	Adding Field Data Transformation Details in a File Request Type on page 979

Adding Services in a File Request Type

Procedure

To define a service in a file request type:

1. Ensure that the **Services** section is expanded when you are defining, editing, or copying a file request type. The **Services** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Sequence	Used to specify the sequence of service invocation and execution.	Yes
Service Type	Used to indicate the type of service. The valid values are: • Business Object • Business Service • Service Script	Yes
Service Name	Used to specify the name of the service to be triggered and executed for each processing record. **Note:** The **Search** (🔍) icon appears corresponding to the **Service Name** field. On clicking the search icon, the **Schema Search** window appears.	Yes
FK Reference	Used to specify the foreign key reference code from where the added or updated details can be viewed. **Note:** The **Search** (🔍) icon appears corresponding to the **FK Reference Code** field. On clicking the search icon, the **Foreign Key Search** window appears.	No
Pre-Processing Algorithm	Used to specify the algorithm required for pre-processing tasks before running the service. **Note:** The **Search** (🔍) icon appears corresponding to the **Pre-Processing Algorithm** field. On clicking the search icon, the **Algorithm Search** window appears.	No
Post Processing Algorithm	Used to specify the algorithm to be executed after a record has been successfully processed. **Note:** The **Search** (🔍) icon appears corresponding to the **Pre-Processing Algorithm** field. On clicking the search icon, the **Algorithm Search** window appears.	No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
Operation	Used to indicate the operation to be performed for the individual service (**Business Object**). The valid values are: • Add • Delete • Read • Replace • Update	No
Dependent Service Name	Used to specify the payload hierarchical relationship and manage dependent services, indicating a parent-child association.	No
Defer Completion	Used to indicate a flag that delays the final status update for a successfully processed request until the life-cycle of the corresponding **Business Object** is complete. **Note:** An API is provided to update the final status of the respective request.	No

2. Enter the required details in the **Services** section.
3. One or more services can be configured. A service can be a child of other service. A service with no inter-dependency can also be configured. If you want to define more than one service for the file request type, click the **Add** (+) icon and then repeat **Step 2**.

> **Note:** However, if you want to remove a service from the file request type, click the **Delete** (🗑) icon corresponding to the service.

4. Click **Save**.
   The service is added in the **File Request Type**.

Related Topics

For more information on...	See...
How to define a file request type	Defining a File Request Type on page 966
How to edit a file request type	Editing a File Request Type on page 981
How to copy a file request type	Copying a File Request Type on page 989

Adding Messages in a File Request Type

Procedure

To add messages in a file request type:

1. Ensure that the **Messages** section is expanded when you are defining, editing, or copying a file request type. The **Messages** section contains the following fields:

Column Name	Column Description	Mandatory (Yes or No)
Message Category	Used to specify the message category number.	Yes

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description	Mandatory (Yes or No)
	Note: The **Search** (🔍) icon appears corresponding to the **Message Category** field. On clicking the search icon, the **Message Category Search** window appears.	
Message Number	Used to specify the number to handle the **Retry** or **Sent for Approval** status.	Yes
	Note: The **Search** (🔍) icon appears corresponding to the **Message Number** field. On clicking the search icon, the **Fetch File Request Type Message Number** window appears.	
Message Text	Used to indicate description of the message.	No
	Note: This is read-only field. As the **Message Number** field is interdependent on **Message Category**, if you search only for **Message Number**, no text will appear in the **Message Text** field. You have to first enter the **Message Category**, and then choose the **Message Number**.	
File Record Status	Used to specify the status for every file record. The valid values are: • Retry • Sent for Approval	Yes

2. Enter the required details in the **Messages** section.
3. One or more messages can be configured. If you want to define more than one message for the file request type, click the **Add** (+) icon and then repeat **Step 2**.

   Note: However, if you want to remove a message from the file request type, click the **Delete** (🗑) icon corresponding to the message.

4. Click **Save**.
   The message is added in the file request type.

Related Topics

For more information on...	See...
How to define a file request type	Defining a File Request Type on page 966
How to edit a file request type	Editing a File Request Type on page 981
How to copy a file request type	Copying a File Request Type on page 989

Adding Header Data Transformation Details in a File Request Type

Prerequisites

To add header data transformation details in a file request type, you should have:

- The **Data Transformation Required** field selected in the **Main** section

Procedure

To add header transformation details in a file request type:

Copyright © 2024, Oracle and/or its affiliates.

1. Click the **Admin** link in the **Actions / Navigation** area.
   A list appears.
2. From the **Admin** menu, select **F** and then click **File Request Type**.
   A sub-menu appears.
3. Click the **Add** option from the **File Request Type** sub-menu.
   The **File Request Type** screen appears.
4. Enter the appropriate details in the **Main** section, if required.
5. Enter the appropriate details in the **Services** and **Messages** section, if required.
6. Ensure that the **Data Transformation** section is expanded when you are defining, editing, or copying a file request type.
7. Ensure **Header Transformation** sub-section is expanded. The **Header Transformation** sub-section contains the following fields:

Column Name	Column Description	Mandatory (Yes or No)
Sequence	Used to specify the mapping of header values in CSV or PSV format files, where the sequence number corresponds to the position of the header in the file.	Yes
Field Name	Used to specify a unique name for the header field that will map to the corresponding file sequence header value. **Note:** A map with these configured **Field Names** and the corresponding **Field Values** is provided as an input to the transformation algorithm.	Yes
Required	Used to indicate header level validations are necessary for every record while uploading data using the **File Transform and Upload (C1-FTRAN)** batch.	No
Edit Input	Used to specify whether the header value can be modified during processing, based on the default value configurations.	No
Default Value	Used to specify default values that will be applied when generating the record payload.	Yes (Conditional) **Note:** This is required when **Sequence Number** is ZERO: 0. The sequence number **ZERO: 0** is reserved for capturing default values.
Date/Time Format	Used to specify the format for date or time values in the input file according to client-defined formats.	No

8. Enter the required header transformation details.
9. One or more transformation details can be configured. If you want to define more than one transformation detail for the file request type, click the **Add** (+) icon and then repeat **Step 7** and **Step 8**.

   Note: However, if you want to remove a transformation detail from the file request type, click the **Delete** (🗑) icon corresponding to the file segment type.

10. Click **Save**.
    The header transformation details are added in the **File Request Type**.

Related Topics

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
File Request Type List screen	File Request Type List on page 960
How to define a file request type	Defining a File Request Type on page 966
How to add services in a file request type	Adding Services in a File Request Type on page 974
How to add messages in a file request type	Adding Messages in a File Request Type on page 975

Adding Footer Data Transformation Details in a File Request Type

Prerequisites

To add footer data transformation details in a file request type, you should have:

- The **Data Transformation Required** field selected in the **Main** section

Procedure

To add footer transformation details in a file request type:

1. Click the **Admin** link in the **Actions / Navigation** area.
   A list appears.
2. From the **Admin** menu, select **F** and then click **File Request Type**.
   A sub-menu appears.
3. Click the **Add** option from the **File Request Type** sub-menu.
   The **File Request Type** screen appears.
4. Enter the appropriate details in the **Main** section, if required.
5. Enter the appropriate details in the **Services** and **Messages** section, if required.
6. Ensure that the **Data Transformation** section is expanded when you are defining, editing, or copying a file request type.
7. Ensure **Footer Transformation** sub-section is expanded. The **Footer Transformation** sub-section contains the following fields:

Column Name	Column Description	Mandatory (Yes or No)
Sequence	Used to specify the mapping of footer values in CSV or PSV format files, where the sequence number corresponds to the position of the footer in the file.	Yes
Field Name	Used to specify a unique name for the footer field that will map to the corresponding file sequence footer value. **Note:** A map with these configured **Field Names** and the corresponding **Field Values** is provided as an input to the transformation algorithm.	Yes
Required	Used to indicate footer level validations are necessary for every record while uploading data using the **File Transform and Upload (C1-FTRAN)** batch.	No
Edit Input	Used to specify whether the footer value can be modified during processing, based on the default value configurations.	No

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description	Mandatory (Yes or No)
Default Value	Used to specify default values that will be applied when generating the record payload.	Yes (Conditional) **Note:** This is required when **Sequence Number** is ZERO: 0. The sequence number **ZERO: 0** is reserved for capturing default values.
Date/Time Format	Used to specify the format for date or time values in the input file according to client-defined formats.	No

8. Enter the required footer transformation details.
9. One or more transformation details can be configured. If you want to define more than one transformation detail for the file request type, click the **Add** (+) icon and then repeat **Step 7** and **Step 8**.

   Note: However, if you want to remove a transformation detail from the file request type, click the **Delete** (🗑) icon corresponding to the file segment type.

10. Click **Save**.
    The footer transformation details are added in the **File Request Type**.

Related Topics

For more information on...	See...
File Request Type List screen	File Request Type List on page 960
How to define a file request type	Defining a File Request Type on page 966
How to add services in a file request type	Adding Services in a File Request Type on page 974
How to add messages in a file request type	Adding Messages in a File Request Type on page 975
How to add header data transformations in a file request type	Adding Header Data Transformation Details in a File Request Type on page 976

Adding Field Data Transformation Details in a File Request Type

Prerequisites

To add field data transformation details in a file request type, you should have:

- The **Data Transformation Required** field selected in the **Main** section

Procedure

To add field transformation details in a file request type:

1. Click the **Admin** link in the **Actions / Navigation** area.
   A list appears.
2. From the **Admin** menu, select **F** and then click **File Request Type**.

Copyright © 2024, Oracle and/or its affiliates.

A sub-menu appears.

3. Click the **Add** option from the **File Request Type** sub-menu.
   The **File Request Type** screen appears.
4. Enter the appropriate details in the **Main** section, if required.
5. Enter the appropriate details in the **Services** and **Messages** section, if required.
6. Ensure that the **Data Transformation** section is expanded when you are defining, editing, or copying a file request type.
7. Ensure **Field Transformation** sub-section is expanded. The **Field Transformation** sub-section contains the following fields:

Column Name	**Column Description**	**Mandatory (Yes or No)**
Sequence	Used to specify the mapping of field values in CSV or PSV format files, where the sequence number corresponds to the position of the fields in the file. **Note:** You can map or configure the same sequence number to multiple fields (Number of Fields can refer to same field value provided in an individual record). A sequence number **ZERO: 0** is reserved for capturing default values.	Yes
Field Name	Used to specify a unique field name, that will contain mapped file sequence field value. **Note:** A map with these configured **Field Names** and the corresponding **Field Values** is provided as an input to the transformation algorithm. If the field data contains the delimiter character, then the each record field should be enclosed with single or double quotes. You need to ensure the correct format for HTML character entities.	Yes
Source Field Path	Used to specify the XPath expression that references elements in XML or JSON files from the legacy system.	No
Map Field XPath	Used to specify an XPath expression syntax to map elements in business objects during transformation. **Note:** The **Search** (🔍) icon appears corresponding to the **Map Field XPath** field. On clicking the search icon, the **Field XPath Search** window appears.	No
Required	Used to indicate field level validations are necessary for every record while uploading data using the **File Transform and Upload (C1-FTRAN)** batch.	No
Record Identifier	Used to indicate whether the records should be uniquely displayed in the file upload dashboard or not.	No
Skip Validation	Used to indicate whether validation should be skipped for length check as per metadata configuration in file upload batch.	No

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description	Mandatory (Yes or No)
Edit Input	Used to specify whether the field value can be modified during processing, based on the default value configurations.	No
Default Value	Used to specify default values that will be applied when generating the record payload.	Yes (Conditional) **Note:** This is required when **Sequence Number** is ZERO: 0. The sequence number **ZERO: 0** is reserved for capturing default values.
Date/Time Format	Used to specify the format of date or time values in the input file according to client-defined formats..	No

8. Enter the required field transformation details.
9. One or more transformation details can be configured. If you want to define more than one transformation detail for the file request type, click the **Add** (+) icon and then repeat **Step 7**and **Step 8**.

> **Note:** However, if you want to remove a transformation detail from the file request type, click the **Delete** (🗑) icon corresponding to the file segment type.

10. Click **Save**.
    The field transformation details are added in the **File Request Type**.

Related Topics

For more information on...	See...
File Request Type List screen	File Request Type List on page 960
How to define a file request type	Defining a File Request Type on page 966
How to add services in a file request type	Adding Services in a File Request Type on page 974
How to add messages in a file request type	Adding Messages in a File Request Type on page 975
How to add header data transformations in a file request type	Adding Header Data Transformation Details in a File Request Type on page 976
How to add footer data transformations in a file request type	Adding Footer Data Transformation Details in a File Request Type on page 978

Editing a File Request Type

Prerequisites

To edit a file request type, you should have:

- File request type business objects defined in the application

Copyright © 2024, Oracle and/or its affiliates.

Procedure

To edit a file request type:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **F** and then click **File Request Type**.
   A sub-menu appears.
3. Click the **Search** option from the **File Request Type** sub-menu.
   The **File Request Type** screen appears.
4. In the **File Request Type List** zone, click the **Edit** (✎) icon in the **Edit** column corresponding to the file request type whose details you want to edit.
   The **File Request Type** screen appears. It contains the following sections:
   - **Main** - Used to specify basic configuration details about the file request type.
   - **Services** - Used to add services in the file request type.
   - **Messages** - Used to specify the messages associated with the file request type.
   - **Data Transformation** - Used to specify the header, footer, and field transformation details in the file request type.

   Note: This section appears only when you select the **Data Transformation Required** field from the **Main** section.

   The **Main** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
File Request Type	Displays the file request type.	Not applicable
Description	Used to specify the description for the file request type. **Tip:** You need to specify the purpose of the file request type.	No
Data Transformation Required	Used to indicate that the type of file or schema which you want to upload is incompatible and needs transformation as per the type of file or schema supported by the application. **Note:** The application supports file type or schema which is in the Extensible Markup Language (XML) format. Therefore, if you are uploading a file or schema in the Extensible Markup Language (XML), you need not select the **Data Transformation Required** field. However, if you are uploading a file or schema which is in a format other than the Extensible Markup Language (XML), you need to select the **Data Transformation Required** field.	Yes (Conditional) **Note:** This field is required while transforming the data received from the external source (i.e. from the customer) as per the format provided in the application.
File Atomicity	Used to indicate whether the system should stop the file upload process and roll back the entire data if any error occurs in any single record of the uploaded file.	No
File Format	Used to indicate the type of file that you want to upload using the file request type. The valid values are: • Comma Separated Values • Extensible Markup Language • Fixed Position • JavaScript Object Notation • Pipe Separated Values	Yes

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	• Tilde Separated Values **Note:** By default, the **Extensible Markup Language** option is selected and the field is disabled. It is enabled only when you select the **Data Transformation Required** field.	
File Extension	Used to specify the extension of the file type selected from the **File Format** list. **Note:** This field is enabled only when you select the **Data Transformation Required** field.	Yes
Upload and Process File Simultaneously	Used to indicate that you want to upload and process the file simultaneously in a single batch run. **Note:** On selecting this field, when the File Transform and Upload (C1-FTRAN) batch is run, the system will upload the files and create a staging table for the records available in the uploaded file. These records will then be assigned to the respective tables of the entity in the database. However, if you do not select this field, the File Transform and Upload (C1-FTRAN) batch will only upload the file and create a staging table for the records available in the uploaded file. To assign the data from the staging table to the respective tables of the entity in the database you need to run the File Request Process (C1-FREQP) batch. This field is selected automatically when the **File Atomicity** field is selected.	No
Root XML Tag	Used to specify the root tag element in the file which you want to upload using the file request type. **Note:** This field is displayed only when the **Data Transformation Required** filed is selected and the **File Format** field is set to **Extensible Markup Language**.	No
File Header Required	Used to indicate whether the file that you want to upload should have header details or not.	No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: If you select this field, the first row data in the uploaded file will be considered as the file header string. If the **Data Transformation Required** filed is selected, you need to configure the header details in the **Header Transformation** sub-section of the **Data Transformation** section. The **Static Header** field is automatically deselected if you select the **File Header Required** field.	
Header XML Tag	Used to specify the header tag element in the file whose data you want to use as the header details. **Note:** This field is displayed only when the **Data Transformation Required**, **File Header Required** fields are selected and the **File Format** field is set to **Extensible Markup Language**.	Yes (Conditional) **Note:** This field is required while uploading an Extensible Markup Language (XML) file for which data transformation is required.
File Footer Required	Used to indicate whether the file that you want to upload should have footer details or not. **Note:** If you select this field, the last row data in the uploaded file will be considered as the file footer string. If the **Data Transformation Required** filed is selected, you need to configure the footer details in the **Footer Transformation** sub-section of the **Data Transformation** section. The **Static Footer** field is automatically deselected if you select the **File Footer Required** field.	No
Footer XML Tag	Used to specify the footer tag element in the file whose data you want to use as the footer details. **Note:** This field is displayed only when the **Data Transformation Required**, **File Footer Required** fields are selected and the **File Format** field is set to **Extensible Markup Language**.	Yes (Conditional) **Note:** This field is required while uploading an Extensible Markup Language (XML) file for which data transformation and footer details is required.
Static Header	Used to indicate that the uploaded file contains a predefined, unchanging header row which does not need dynamic validation during processing.	No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: If the **Data Transformation Required** filed is selected, you need to configure the static header details in the **Header Transformation** sub-section of the **Data Transformation** section. The **File Header Required** field is automatically deselected if you select the **Static Header** field.	
Static Footer	Used to indicate that the uploaded file contains a predefined, unchanging footer row typically used for summary or ending markers without requiring dynamic validation during processing.	No
	Note: If the **Data Transformation Required** filed is selected, you need to configure the static footer details in the **Footer Transformation** sub-section of the **Data Transformation** section. The **File Footer Required** field is automatically deselected if you select the **Static Footer** field.	
Service Log Required	Used to indicate whether you want to log details about the creation and updates made in every record of the entity corresponding to the services configured for the file request type.	No
	Note: The service log details are used on the **File Request dashboard** screen for navigating to the changes made in the corresponding entity.	
Skip Duplicates	Used to indicate whether you want to skip the execution of duplicate records in the uploaded file.	No
	Note: If you select this option, none of the duplicate records will be processed. For Example: If you have two duplicate records in the uploaded file and the **Skip Duplicate** field is selected, then, the system will skip both the records while processing the file. Duplicate entries are identified when you select the **Record Identifier** field corresponding to the **Field Name** configured in the **Field Transformation** sub-section of the **Data Transformation** section.	
File Validation Algorithm	Used to specify the algorithm which you want to use for validating the data in the uploaded file. The valid values are:	Yes (Conditional)

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	• **File Request Header Footer Validation** (C1-FRHFVA) - Used to validate the header and footer provided in the file. If the **File Header Required** or **Static Header** field is selected, the algorithm checks whether the header line in the file is marked with the header marker **H**. If the **File Footer Required** or **Static Footer** field is selected, the algorithm checks whether the footer line in the file is marked with the footer marker **T**, followed by the count of records in the file. **Note:** If none of the above mentioned conditions are met, the system will stop processing the file. • **Sample File Validation** (C1-FRHVA) - Used to check whether the associated algorithm is invoked and logs the fields available in the header and footer while processing the file. **Note:** The **Search** (🔍) icon appears corresponding to the **File Validation Algorithm** field. On clicking the search icon, the **Algorithm Search** search window appears where you can search for the respective algorithm. The above algorithms are shipped with the product. The implementation team can create custom algorithm type for this algorithm entity based on the customer requirements.	**Note:** This field is required for validating the header and footer information while processing the data provided in the file using the file request type.
Data Transformation Algorithm	Used to indicate the algorithm using which the data from the uploaded file is transformed into data which is compliant with the schema defined for the business service in the application. **Note:** The **Search** (🔍) icon appears corresponding to the **Data Transformation Algorithm** field. On clicking the search icon, the **Algorithm Search** search window appears where you can search for the respective algorithm. This field appears only when you select the **Data Transformation Required** field.	Yes (Conditional) **Note:** This field is required while data transformation.
Maximum Retry for Error Record	Used to specify the maximum number of retry attempts that are allowed for records which are in the error status while processing the uploaded file.	No
Validate Record Payload	Used to indicate that you want to validate and process only those records that have payloads for other services that are configured using the file request type.	No
Display Profile	Used to indicate how the date should be displayed when the entity is created. The valid values are:	No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	• Europe • North America • HIJRI Format • Other	
File Upload Approval Required	Used to indicate whether approval is required to upload and process the file using the file request type. **Note:** If you select this option you need to ensure that the approval workflow for file upload interface is configured in the system. When the File Transform and Upload (C1-FTRAN) batch is run, the system will create an Approval Transaction ID and mark the file in the **Pending for Approval** status. The approver can then search and approve the file from the **Approval Transaction** screen using the Approval Transaction ID. Once the approver approves the file from the **Approval Transaction** screen, the File Transform and Upload (C1-FTRAN) batch is triggered automatically for processing the file.	No
Date Format	Used to specify the format in which you want to display the date. **Note:** This field appears only when the **Other** option is selected from the **Display Profile** list.	No
Time Format	Used to specify the format in which you want to display the time. **Note:** This field appears only when the **Other** option is selected from the **Display Profile** list.	No
File Decryption Required	Used to indicate whether decryption is required for the file before processing. **Note:** If you select this field, the system will check for the encryption details configured for the file upload interface in the **Master Configuration** screen.	No
Duplicate Record Identifier Expression	Used to specify the field for which you want to identify duplicate records in the uploaded file.	No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: If you want to specify multiple fields, you can do so by adding <space>&&<space> as the separator in between the fields. If you want to specify a combination of fields, you can do so by adding <space>||<space> as the separator in between the fields.	
Group Duplicate Records	Used to indicate whether duplicate records in the uploaded files should be processed in a same thread or not.	No
Duplicate Records Order By	Used to indicate the order in which you want to process the duplicate records.	No
External System	Used to specify an external system for configuring a file request type.	No
Force File Archival	Used to indicate whether the file should be archived and moved to the location mentioned in the **Master Configuration** screen after processing.	No
Record Size Greater Than 32KB	Used to indicate whether each record in the file which you want to upload using the file request type is greater than 32 KB or not. **Note:** Record size greater than 32 KB will be stored in the **CLOB** data type. Whereas, record size less than 32 KB, will be stored in **VARCHAR2** data type in the database.	No
Error To Do Type	Used to indicate the To Do entry of the specified To Do Type must be created in case of any error encountered while processing the file using the file request type. **Note:** The **Search** (🔍) icon appears corresponding to the **Error To Do Type** field. On clicking the search icon, the **To Do Type Details** window appears where you can search for the respective To Do Type.	No
Detailed Description	Used to specify additional information about the file request type.	No

5. Modify the required details in the main section.
6. Add, edit, or delete services from the file request type, if required.
7. Add, edit, or delete messages from the file request type, if required.
8. Add, edit, or delete the data transformation details from file request type, if required.
9. Click **Save**.
   The changes made to the file request type are saved.

Related Topics

For more information on...	See...
File Request Type screen	File Request Type on page 959

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
File Request Type List zone	File Request Type List on page 960
How to add services in a file request type	Adding Services in a File Request Type on page 974
How to add messages in a file request type	Adding Messages in a File Request Type on page 975
How to add header data transformation details in a file request type	Adding Header Data Transformation Details in a File Request Type on page 976
How to add footer data transformation details in a file request type	Adding Footer Data Transformation Details in a File Request Type on page 978
How to add field data transformation details in a file request type	Adding Field Data Transformation Details in a File Request Type on page 979

Deleting a File Request Type

Procedure

To delete a file request type:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **F** and then click **File Request Type**.
   A sub-menu appears.
3. Click the **Search** option from the **File Request Type** sub-menu.
   The **File Request Type** screen appears.
4. In the **File Request Type List** zone, click the **Delete** (🗑) icon in the **Delete** column corresponding to the file request type that you want to delete.
   A message appears confirming whether you want to delete the file request type.
5. Click **OK**.
   The file request type is deleted.

Note: You can delete a file request type only when a file upload request is not yet created using the file request type.

Related Topics

For more information on...	See...
File Request Type screen	File Request Type on page 959
File Request Type List zone	File Request Type List on page 960

Copying a File Request Type

Instead of creating a file request type from scratch, you can create a new file request type using an existing file request type. This is possible by copying a file request type. On copying a file request type, the details including the services, messages, and the data transformation details also get copied to the new file request type. You can then edit the details, if required.

Prerequisites

To copy a file request type, you should have:

- File request type (whose copy you want to create) defined in the application

Copyright © 2024, Oracle and/or its affiliates.

- File request type business objects defined in the application

Procedure

To copy a file request type:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **F** and then click **File Request Type**.
   A sub-menu appears.
3. Click the **Search** option from the **File Request Type** sub-menu.
   The **File Request Type** screen appears.
4. In the **File Request Type List** zone, click the **Duplicate** (⧉) icon in the **Duplicate** column corresponding to the file request type whose copy you want to create.
   The **File Request Type** screen appears. It contains the following sections:
   - **Main** - Used to specify basic configuration details about the file request type.
   - **Services** - Used to add services in the file request type.
   - **Messages** - Used to specify the messages associated with the file request type.
   - **Data Transformation** - Used to specify the header, footer, and field transformation details in the file request type.

   Note: This section appears only when you select the **Data Transformation Required** field from the **Main** section.

   The **Main** section contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
File Request Type	Used to specify the file request type.	Yes
Description	Used to specify the description for the file request type. **Tip:** You need to specify the purpose of the file request type.	No
Data Transformation Required	Used to indicate that the type of file or schema which you want to upload is incompatible and needs transformation as per the type of file or schema supported by the application. **Note:** The application supports file type or schema which is in the Extensible Markup Language (XML) format. Therefore, if you are uploading a file or schema in the Extensible Markup Language (XML), you need not select the **Data Transformation Required** field. However, if you are uploading a file or schema which is in a format other than the Extensible Markup Language (XML), you need to select the **Data Transformation Required** field.	Yes (Conditional) **Note:** This field is required while transforming the data received from the external source (i.e. from the customer) as per the format provided in the application.
File Atomicity	Used to indicate whether the system should stop the file upload process and roll back the entire data if any error occurs in any single record of the uploaded file.	No
File Format	Used to indicate the type of file that you want to upload using the file request type. The valid values are: • Comma Separated Values • Extensible Markup Language • Fixed Position • JavaScript Object Notation	Yes

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	• Pipe Separated Values • Tilde Separated Values **Note:** By default, the **Extensible Markup Language** option is selected and the field is disabled. It is enabled only when you select the **Data Transformation Required** field.	
File Extension	Used to specify the extension of the file type selected from the **File Format** list. **Note:** This field is enabled only when you select the **Data Transformation Required** field.	Yes
Upload and Process File Simultaneously	Used to indicate that you want to upload and process the file simultaneously in a single batch run. **Note:** On selecting this field, when the File Transform and Upload (C1-FTRAN) batch is run, the system will upload the files and create a staging table for the records available in the uploaded file. These records will then be assigned to the respective tables of the entity in the database. However, if you do not select this field, the File Transform and Upload (C1-FTRAN) batch will only upload the file and create a staging table for the records available in the uploaded file. To assign the data from the staging table to the respective tables of the entity in the database you need to run the File Request Process (C1-FREQP) batch. This field is selected automatically when the **File Atomicity** field is selected.	No
Root XML Tag	Used to specify the root tag element in the file which you want to upload using the file request type. **Note:** This field is displayed only when the **Data Transformation Required** filed is selected and the **File Format** field is set to **Extensible Markup Language**.	No
File Header Required	Used to indicate whether the file that you want to upload should have header details or not.	No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: If you select this field, the first row data in the uploaded file will be considered as the file header string. If the **Data Transformation Required** filed is selected, you need to configure the header details in the **Header Transformation** sub-section of the **Data Transformation** section. The **Static Header** field is automatically deselected if you select the **File Header Required** field.	
Header XML Tag	Used to specify the header tag element in the file whose data you want to use as the header details. **Note:** This field is displayed only when the **Data Transformation Required**, **File Header Required** fields are selected and the **File Format** field is set to **Extensible Markup Language**.	Yes (Conditional) **Note:** This field is required while uploading an Extensible Markup Language (XML) file for which data transformation is required.
File Footer Required	Used to indicate whether the file that you want to upload should have footer details or not. **Note:** If you select this field, the last row data in the uploaded file will be considered as the file footer string. If the **Data Transformation Required** filed is selected, you need to configure the footer details in the **Footer Transformation** sub-section of the **Data Transformation** section. The **Static Footer** field is automatically deselected if you select the **File Footer Required** field.	No
Footer XML Tag	Used to specify the footer tag element in the file whose data you want to use as the footer details. **Note:** This field is displayed only when the **Data Transformation Required**, **File Footer Required** fields are selected and the **File Format** field is set to **Extensible Markup Language**.	Yes (Conditional) **Note:** This field is required while uploading an Extensible Markup Language (XML) file for which data transformation and footer details is required.
Static Header	Used to indicate that the uploaded file contains a predefined, unchanging header row which does not need dynamic validation during processing.	No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: If the **Data Transformation Required** filed is selected, you need to configure the static header details in the **Header Transformation** sub-section of the **Data Transformation** section. The **File Header Required** field is automatically deselected if you select the **Static Header** field.	
Static Footer	Used to indicate that the uploaded file contains a predefined, unchanging footer row typically used for summary or ending markers without requiring dynamic validation during processing.	No
	Note: If the **Data Transformation Required** filed is selected, you need to configure the static footer details in the **Footer Transformation** sub-section of the **Data Transformation** section. The **File Footer Required** field is automatically deselected if you select the **Static Footer** field.	
Service Log Required	Used to indicate whether you want to log details about the creation and updates made in every record of the entity corresponding to the services configured for the file request type.	No
	Note: The service log details are used on the **File Request dashboard** screen for navigating to the changes made in the corresponding entity.	
Skip Duplicates	Used to indicate whether you want to skip the execution of duplicate records in the uploaded file.	No
	Note: If you select this option, none of the duplicate records will be processed. For Example: If you have two duplicate records in the uploaded file and the **Skip Duplicate** field is selected, then, the system will skip both the records while processing the file. Duplicate entries are identified when you select the **Record Identifier** field corresponding to the **Field Name** configured in the **Field Transformation** sub-section of the **Data Transformation** section.	
File Validation Algorithm	Used to specify the algorithm which you want to use for validating the data in the uploaded file. The valid values are:	Yes (Conditional)

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	• **File Request Header Footer Validation** (C1-FRHFVA) - Used to validate the header and footer provided in the file. If the **File Header Required** or **Static Header** field is selected, the algorithm checks whether the header line in the file is marked with the header marker **H**. If the **File Footer Required** or **Static Footer** field is selected, the algorithm checks whether the footer line in the file is marked with the footer marker **T**, followed by the count of records in the file. **Note:** If none of the above mentioned conditions are met, the system will stop processing the file. • **Sample File Validation** (C1-FRHVA) - Used to check whether the associated algorithm is invoked and logs the fields available in the header and footer while processing the file. **Note:** The **Search** (🔍) icon appears corresponding to the **File Validation Algorithm** field. On clicking the search icon, the **Algorithm Search** search window appears where you can search for the respective algorithm. The above algorithms are shipped with the product. The implementation team can create custom algorithm type for this algorithm entity based on the customer requirements.	**Note:** This field is required for validating the header and footer information while processing the data provided in the file using the file request type.
Data Transformation Algorithm	Used to indicate the algorithm using which the data from the uploaded file is transformed into data which is compliant with the schema defined for the business service in the application. **Note:** The **Search** (🔍) icon appears corresponding to the **Data Transformation Algorithm** field. On clicking the search icon, the **Algorithm Search** search window appears where you can search for the respective algorithm. This field appears only when you select the **Data Transformation Required** field.	Yes (Conditional) **Note:** This field is required while data transformation.
Maximum Retry for Error Record	Used to specify the maximum number of retry attempts that are allowed for records which are in the error status while processing the uploaded file.	No
Validate Record Payload	Used to indicate that you want to validate and process only those records that have payloads for other services that are configured using the file request type.	No
Display Profile	Used to indicate how the date should be displayed when the entity is created. The valid values are:	No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	• Europe • North America • HIJRI Format • Other	
File Upload Approval Required	Used to indicate whether approval is required to upload and process the file using the file request type. **Note:** If you select this option you need to ensure that the approval workflow for file upload interface is configured in the system. When the File Transform and Upload (C1-FTRAN) batch is run, the system will create an Approval Transaction ID and mark the file in the **Pending for Approval** status. The approver can then search and approve the file from the **Approval Transaction** screen using the Approval Transaction ID. Once the approver approves the file from the **Approval Transaction** screen, the File Transform and Upload (C1-FTRAN) batch is triggered automatically for processing the file.	No
Date Format	Used to specify the format in which you want to display the date. **Note:** This field appears only when the **Other** option is selected from the **Display Profile** list.	No
Time Format	Used to specify the format in which you want to display the time. **Note:** This field appears only when the **Other** option is selected from the **Display Profile** list.	No
File Decryption Required	Used to indicate whether decryption is required for the file before processing. **Note:** If you select this field, the system will check for the encryption details configured for the file upload interface in the **Master Configuration** screen.	No
Duplicate Record Identifier Expression	Used to specify the field for which you want to identify duplicate records in the uploaded file.	No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: If you want to specify multiple fields, you can do so by adding <space>&&<space> as the separator in between the fields. If you want to specify a combination of fields, you can do so by adding <space>||<space> as the separator in between the fields.	
Group Duplicate Records	Used to indicate whether duplicate records in the uploaded files should be processed in a same thread or not.	No
Duplicate Records Order By	Used to indicate the order in which you want to process the duplicate records.	No
External System	Used to specify an external system for configuring a file request type.	No
Force File Archival	Used to indicate whether the file should be archived and moved to the location mentioned in the **Master Configuration** screen after processing.	No
Record Size Greater Than 32KB	Used to indicate whether each record in the file which you want to upload using the file request type is greater than 32 KB or not. **Note:** Record size greater than 32 KB will be stored in the **CLOB** data type. Whereas, record size less than 32 KB, will be stored in **VARCHAR2** data type in the database.	No
Error To Do Type	Used to indicate the To Do entry of the specified To Do Type must be created in case of any error encountered while processing the file using the file request type. **Note:** The **Search** (🔍) icon appears corresponding to the **Error To Do Type** field. On clicking the search icon, the **To Do Type Details** window appears where you can search for the respective To Do Type.	No
Detailed Description	Used to specify additional information about the file request type.	No

5. Enter the required details in the **Main** section.
6. Add, edit, or delete services from the file request type, if required.
7. Add, edit, or delete messages from the file request type, if required.
8. Add, edit, or delete the data transformation details from file request type, if required.
9. Click **Save**.
   The new file request type is defined.

Related Topics

For more information on...	See...
File Request Type screen	File Request Type on page 959

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
File Request Type List zone	File Request Type List on page 960
How to add services in a file request type	Adding Services in a File Request Type on page 974
How to add messages in a file request type	Adding Messages in a File Request Type on page 975
How to add header data transformation details in a file request type	Adding Header Data Transformation Details in a File Request Type on page 976
How to add footer data transformation details in a file request type	Adding Footer Data Transformation Details in a File Request Type on page 978
How to add field data transformation details in a file request type	Adding Field Data Transformation Details in a File Request Type on page 979

Viewing the File Request Type Details

Procedure

To view the details of a file request type:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **F** and then click **File Request Type**.
   A sub-menu appears.
3. Click the **Search** option from the **File Request Type** sub-menu.
   The **File Request Type** screen appears.
4. In the **File Request Type List** zone, click the **Broadcast** ( ) icon corresponding to the **File Request Type** column, whose details you want to view.
   The **File Request Type** zone appears.
5. View the details of the file request type in the **File Request Type** zone.

Related Topics

For more information on...	See...
File Request Type screen	File Request Type on page 959
File Request Type List zone	File Request Type List on page 960
File Request Type zone	File Request Type on page 960

File Upload Dashboard

The **File Upload Dashboard** screen allows you to:

- Search for a file
- Mark the files for archival
- View the details of a file

Copyright © 2024, Oracle and/or its affiliates.

The **File Upload Dashboard** screen contains the following zones:

- Search File on page 998
- File Upload Failure Reason on page 1000
- Successfully Uploaded Record Summary on page 1001
- File Record Status Summary on page 1002
- File Error Messages List on page 1002
- File Header Detail on page 1003
- File Footer Detail on page 1003

Search File

The **Search File** zone allows you to search for a file using various search criteria. It contains the following two sections:

- **Search Criteria** - The **Search Criteria** section contains the following fields:

Field Name	Field Description
File Name	Used to search for files having a particular name.
File Status	Used to search for files which are in a particular status. The valid values are: • Approval Pending • Complete • Error • Pending • Reject
File Request Type	Used to search for files which are uploaded using a particular file request type.
External System	Used to search for files which are uploaded from an external system.
File Business From Date	Used to search for files from the legacy system that are uploaded on or after a particular date.
File Business To Date	Used to search for files from the legacy system that are uploaded on or before a particular date.
File Upload From Date	Used to search for files that are uploaded on or after a particular date.
File Upload To Date	Used to search for files that are uploaded on or before a particular date.
Ready for Archival	Used to search for files which are ready to be archived.
File ID	Used to search for a specific file.

- **Search Results** - On clicking the **Search** button, the search results appear based on the specified search criteria. The **Search Results** section contains the following columns:

Column Name	Column Description
File ID	Displays the unique identifier assigned to each uploaded file.
File Name	Displays the name of the file.
File Request Type	Displays the file request type using which the file is uploaded in the system.
	Note: It has a link. On clicking the link, the **File Request Type** screen appears with the details of the respective file request type.

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
Status	Displays the current status of the file. The valid values are: • Approval Pending • Complete • Error • Pending • Reject
Go To Batch Run Tree	Displays the batch and batch run number associated with the file.
	Note: It has a link. On clicking the link, the **Batch Run Tree** screen appears with the details of the executed batch.
Pending Records	Displays the number of records which are in the **Pending** status for a particular file.
	Note: It has a link. On clicking the link, the **File Record Details** screen appears with the details of the records which are in the **Pending** status.
Processed Records	Displays the number of records which are in the **Processed** status for a particular file.
	Note: It has a link. On clicking the link, the **File Record Details** screen appears with the details of the records which are in the **Processed** status.
Error Records	Displays the number of records which are in the **Error** status for a particular file.
	Note: It has a link. On clicking the link, the **File Record Details** screen appears with the details of the records which are in the **Error** status.
Retry Records	Displays the number of records which are in the **Retry** status for a particular file.
	Note: It has a link. On clicking the link, the **File Record Details** screen appears with the details of the records in the **Retry** status.
Ignore Records	Displays the number of records which are in the **Ignore** status for a particular file.
	Note: It has a link. On clicking the link, the **File Record Details** screen appears with the details of the records in the **Ignore** status.
Skipped Records	Displays the number of records which are in the **Skipped** status for a particular file.
	Note: It has a link. On clicking the link, the **File Record Details** screen appears with the details of the records in the **Skipped** status.
In Progress Records	Displays the number of records which are in the **In Progress** status for a particular file.
	Note: It has a link. On clicking the link, the **File Record Details** screen appears with the details of the records in the **In Progress** status.
Records with Retry Limit Exceeded	Displays the number of records which are in the **Retry Limit Exceeded** status for a particular file..

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
	Note: It has a link. On clicking the link, the **File Record Details** screen appears with the details of the records in the **Retry Limit Exceeded** status.
Total Records	Displays the total number of records in the file.
	Note: It has a link. On clicking the link, the **File Record Details** screen appears with the details of the records in the file.
Records Marked for Archival	Displays the total number of records selected for archival in the file.
File Business Date	Displays the date on which the file was uploaded from the legacy system.
File Upload Date Time	Displays the date and time on which the file was uploaded or staged in the system.

In addition, the **Search Results** section contains the following button:

Button Name	Button Description
File Archival Update	Used to mark or unmark the selected files for archival.

You can filter the list using various search criteria (such as, **File Name**, **File Status**, **File Request Type**, **External System**, **File Business From Date**, **File Business To Date**, **File Upload From Date**, **File Upload To Date**, **Ready for Archival**, and **File ID**) available in the **Filter** area. By default, the **Filter** area is hidden. You can view the **Filter** area by clicking the **Filters** (▽) icon in the upper right corner of this zone or by clicking the **Expand Filters** button available in the **Search Results** section.

On clicking the **Broadcast** (ᯤ) icon corresponding to a file, various zones (such as, **File Upload Failure Reason**, **Successfully Uploaded Record Summary**, **File Record Status Summary**, **File Error Messages List**, **File Header Detail**, and **File Footer Detail**) appears with the details related to the respective file.

Note: Pagination is used to display limited number of records in this zone. You can use the navigation links, such as **Previous** and **Next** to navigate between pages.

Related Topics

For more information on...	See...
Search File zone	Search File on page 998
File Upload Failure Reason zone	File Upload Failure Reason on page 1000
Successfully Uploaded Record Summary zone	Successfully Uploaded Record Summary on page 1001
File Record Status Summary zone	File Record Status Summary on page 1002
File Error Messages List zone	File Error Messages List on page 1002
File Header Detail zone	File Header Detail on page 1003
File Footer Detail zone	File Footer Detail on page 1003

File Upload Failure Reason

The **File Upload Failure Reason** zone provides critical details regarding the failures encountered during the file upload process. It contains the following fields:

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
Error Message	Displays the message for the failure in file upload process.
File Business Date	Displays the business date associated with the file that failed to upload.
File Upload Date Time	Displays the exact time-stamp when the file was uploaded.

By default, the **File Upload Failure Reason** zone does not appear in the **File Upload Dashboard** screen. It appears when you click the **Broadcast** (📡) icon corresponding to the file in the search result of **Search File** zone.

Related Topics

For more information on...	See...
File Upload Dashboard screen	File Upload Dashboard on page 997
Search File zone	Search File on page 998

Successfully Uploaded Record Summary

The **Successfully Uploaded Record Summary** zone provides an overview of records that were successfully uploaded during the file upload process. It contains the following fields:

Column Name	Column Description
Go To Batch Run Tree	Displays the batch code and batch number which are associated with the search file record.
	Note: It has a link. On clicking the link, the **Batch Run Tree** screen appears which allows you to view the status of a specific execution of a batch job.
Number of records Added	Displays the count of new records added to the system through the upload process.
	Note: It has a link. On clicking the link, the **File Record Details** screen appears with the details of the records in the added in the ORMB.
Number of records updated	Displays the count of records that were updated during the upload process.
	Note: It has a link. On clicking the link, the **File Record Details** screen appears with the details of the records in the **Processed** status and update in operation.

By default, the **Successfully Uploaded Record Summary** zone does not appear in the **File Upload Dashboard** screen. It appears when you click the **Broadcast** (📡) icon corresponding to the file in the **Search Results** section of **Search File** zone.

Related Topics

For more information on...	See...
File Upload Dashboard screen	File Upload Dashboard on page 997
Search File zone	Search File on page 998

Copyright © 2024, Oracle and/or its affiliates.

File Record Status Summary

The **File Record Status Summary** zone provides an overview of the files categorized by their respective record statuses. It allows users to quickly identify and monitor the status distribution of records processed through the file upload system. It contains the following fields:

Column Name	Column Description
Record status	Displays the status of the records.
Number of records	Displays the total count of records corresponding to each status.

By default, the **Successfully Uploaded Record Summary** zone does not appear in the **File Upload Dashboard** screen. It appears when you click the **Broadcast** (📶) icon corresponding to the file in the **Search Results** section of **Search File** zone.

Related Topics

For more information on...	See...
File Request Message Detail zone	File Request Message Detail
File Request Search zone	Search File on page 998

File Error Messages List

The **File Records Error Messages List** zone displays details about errors that occur in file records during the file upload process. It contains the following fields:

Column Name	Column Description
Message Category	Displays the category of the error message.
Message Number	Displays the message number associated with the error message for reference.
Retry Records	Displays the count of records in the **Retry** status.
	Note: It has a link. On clicking the link, the **File Record Detail** screen appears with the details of the records in the **Retry** status.
Records With Retry Limit Exceeded	Displays the count of records for which the retry attempts have exceeded the allowable limit.
	Note: It has a link. On clicking the link, the **File Record Detail** screen appears with the details of the records in the **Retry Limit Exceed** status.
Error Records	Displays the count of records in the **Error** status.
	Note: It has a link. On clicking the link, the **File Record Detail** screen appears with the details of the records in the **Error** status.

By default, the **File Error Messages List** zone does not appear in the **File Upload Dashboard** screen. It appears when you click the **Broadcast** (📶) icon corresponding to the file in the **Search Results** section of **Search File** zone.

Related Topics

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
File Upload Dashboard screen	File Upload Dashboard on page 997
Search File zone	Search File on page 998

File Header Detail

The **File Header Detail** zone displays detailed information about the header segment of the uploaded file. This zone highlights both the specific field names and their corresponding values as captured in the transformation process. It contains the following fields:

Column Name	Column Description
Header	Displays the field name of the respective header segment type, as defined in the **Transformation Details** section.
Value	Displays the value corresponding to the respective header segment type.
	Note: This column shows both the **Header Values** specified in the file and the default value specified in the **Transformation Details**.

By default, the **File Header Detail** zone does not appear in the **File Upload Dashboard** screen. It appears when you click the **Broadcast** (📡) icon corresponding to the file in the **Search Results** section of **Search File** zone.

Related Topics

For more information on...	See...
File Upload Dashboard screen	File Upload Dashboard on page 997
Search File zone	Search File on page 998

File Footer Detail

The **File Footer Detail** zone displays detailed information about the footer segment of the uploaded file. This zone highlights both the specific field names and their corresponding values as captured in the data transformation process. It contains the following fields:

Column Name	Column Description
Footer	Displays the field name of respective footer segment type as defined in the **Transformation Details** section.
Value	Displays the value corresponding to the respective footer segment type.
	Note: This column will show both, **Footer values** specified in the file and the default value as specified in the **Transformation Details** section.

By default, the **File Footer Detail** zone does not appear in the **File Upload Dashboard** screen. It appears when you click the **Broadcast** (📡) icon corresponding to the file in the **Search Results** section of **Search File** zone.

Related Topics

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
File Upload Dashboard screen	File Upload Dashboard on page 997
Search File zone	Search File on page 998

Searching for File

Prerequisites

To search for a file, you should have:

- File requests defined in the application.

Procedure

To search for a file:

1. Click the **Menu** link in the **Application** toolbar.
   A list appears.
2. From the **Main** menu, select **Integration** and then click **File Upload Dashboard**.
   The **File Upload Dashboard** screen appears.
3. Enter the search criteria in the **Search File** zone.

   Note:

   ORMB search engine supports wildcard search, where you can substitute the percentage (%) symbol as a stand in for any word or letter in a search criteria. You can use the '%' wildcard character in all input fields except the date and ID fields. The '%' wildcard character is suffixed automatically at the end of the partial search criteria. Therefore, you may or may not specify the wildcard character at the end of the partial search criteria. However, you have to prefix the wildcard character manually wherever required.

   You must specify either the **File Name** or **File Request Type** as the search criterion while searching for a file request.

4. Click **Search**.
   A list of file requests that meet the search criteria appears in the **Search Results** section.

Related Topics

For more information on...	See...
File Upload Dashboard screen	File Upload Dashboard on page 997
Search File zone	Search File on page 998

Marking a File for Archival

Procedure

To mark a file for archival:

1. Search for the file in the **Search File** zone of the **File Upload Dashboard** screen.
2. In the **Search Results** section, select the check-box corresponding to the file that you want to archive.
3. Click the **File Archival Update** button.
   The **File Upload Data Archival Tag Update** window appears.
4. Select the **Y** option from the **Mark Data Archival Flag** field.

   Note: If you want to unmark the file from archival, select the **N** option from the **Mark Data Archival Flag** field.

Copyright © 2024, Oracle and/or its affiliates.

5. Click **OK**.
   The selected file is marked for archival.

Related Topics

For more information on...	See...
File Upload Dashboard screen	File Upload Dashboard on page 997
Search File zone	Search File on page 998

Viewing the Details of a File

Procedure

To view the details of a file:

1. Search for the file in the **Search File** zone of the **File Upload Dashboard** screen.
2. In the **Search Results** section, click the **Broadcast** ( ) icon corresponding to the file whose details you want to view.
   The following zones appear:
   - File Upload Failure Reason on page 1000
   - Successfully Uploaded Record Summary on page 1001
   - File Record Status Summary on page 1002
   - File Error Messages List on page 1002
   - File Header Detail on page 1003
   - File Footer Detail on page 1003
3. View the specific details of the file in the respective zone.

Related Topics

For more information on...	See...
File Upload Dashboard screen	File Upload Dashboard on page 997
Search File zone	Search File on page 998

File Record Details

The **File Record Details** screen allows you to:

- Search for a File Record
- View the Details of a File Record
- Update the Status of a File Record
- Mark the file record(s) for archival
- View the record payload details
- View the service payload details
- View the transformed payload details

The **File Record Details** screen contains the following zones:

- File Records on page 1006
- File Record Service Details on page 1008
- File Record Identifier Details on page 1008
- File Record Status Transition on page 1009
- File Record Error Message History on page 1009

Copyright © 2024, Oracle and/or its affiliates.

File Records

The **File Records** zone enables to search for the file records using various criteria. It contains the following two sections:

- **Search Criteria** - The **Search Criteria** section contains the following fields:

Field Name	Field Description
Record Identifier1	Used to search for records based on their primary or unique identification.
Record Identifier2	Used to search for records using a secondary identifier or supporting reference, providing more precision with alternate IDs like customer IDs or account numbers.
Record Identifier3	Used to search for records with an additional layer of identification, refining searches using details such as versions or extended attributes.
Record Identifier4	Used to search for records using another record identification attribute or related metadata, such as, department codes, categories, or tags.
Record Identifier5	Used to search for records with an auxiliary or fifth-level identifier, offering extended identification for specific records like groups or processing flags.
Operation	Used to search for records based on the associated action or operation. The valid values are: • Add • Delete • Read • Replace • Update
Batch Control	Used to search for records using the control number or code assigned to a specific batch of records.
Batch Number	Used to search for records by a specific batch number to filter grouped records.
Record Status	Used to search for records based on the particular status. The valid values are: • Error • Ignore • In progress • Pending • Processed • Retry • Retry Limit Exceeded • Sent For Approval • Skipped
Ready for Archival	Used to search for records flagged as ready for archival.

- **Search Results** - On clicking the **Search** button, the search results appear based on the specified search criteria. The **Search Results** section contains the following columns:

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
File Record Id	Displays the unique identifier that represents the individual file record.
File Record Identifier1	Displays the field and the value that is defined as the record identifier while defining the configuration for the file upload request. **Note:** The data in the **Record Identifier** column(s) appears when the uploaded file is in the **Error** status.
File Record Identifier2	Displays a secondary identifier or supporting reference for the record. It adds more precision by including alternate IDs like customer IDs or account numbers.
File Record Identifier3	Displays an additional layer of identification for extended search criteria. It helps to refine searches using further details such as versions or extended attributes.
File Record Identifier4	Displays another record identification attribute or related metadata. It filter records based on secondary details like department codes, categories, or tags.
File Record Identifier5	Displays an auxiliary or fifth-level identifier for additional precision. It provides extended identification for specific records, like groups or processing flags.
Error Message	Displays the file record error message.
Ready for Archival	Indicates whether the record is flagged as **Ready for Archival**.
Record Payload	Displays the actual record payload that is uploaded by the legacy system. **Note:** On clicking the **View** (🔗) icon in the **Record Payload** column, the **Record Payload** window appears where you can view the details of the record payload.
Transformed Payload	Displays the XML-transformed version of the record payload. **Note:** On clicking the **View** (🔗) icon in the **Transformed Payload** column, the **Transformed Payload** window appears where you can view the details of the transformed records. The transformed payload in XML format is applicable to those file upload requests that have **Data Transformation Required** flag set to **True**.
Upload Date Time	Displays the date and time when the record was uploaded.
Last Update Date Time	Displays the date and time when the record was last updated.

In addition to the above columns, the **Search Results** section contains the following buttons:

Button Name	Button Description
Update Record Status	Used to update the status of the records.
Record Archival Update	Used to archive the marked records.

You can filter the list using various search criteria (such as, **Record Identifier1**, **Record Identifier2**, **Record Identifier3**, **Record Identifier4**, **Record Identifier5**, **Operation**, **Batch Control**, **Batch Number**, **Record Status** and **Ready for Archival**) available in the **Filter** area. By default, the **Filter** area is hidden. You can view the **Filter** area by clicking the **Filters** (▽) icon in the upper right corner of this zone or by clicking the **Expand Filter** button available in the **Search Results** section.

Copyright © 2024, Oracle and/or its affiliates.

On clicking the **Broadcast** (📶) icon corresponding to a file record, the various zones (such as, **File Record Service Details**, **File Record Identifier Details**, **File Record Status Transition**, and **File Record Error Message History**) appears with the details related to a respective record.

Related Topics

For more information on...	See...
File Record Service Detail zone	File Record Service Details on page 1008
File Record Identifier Detail zone	File Record Identifier Details on page 1008
File Record Status Transition zone	File Record Status Transition on page 1009
File Record Error Message History zone	File Record Error Message History on page 1009

File Record Service Details

The **File Record Service Details** zone provides detailed information about the execution of services for successfully processed records. It contains the following columns:

Column Name	Column Description
Sequence	Displays the sequence number of service execution for the file record. This sequence ensures the correct order of execution for dependent services.
Service Name	Displays the name of the service invoked for processing the record.
Entity Information	Displays the entity-related information associated with the service.
	Note: It has a link. On clicking the link, the associated **Service** screen appears where you can view the details of the respective entity. Example -**Bill Route Type** screen.
Service Payload	Displays the payload passed to the service for processing the file record.
	Note: On clicking the **View** (🔗) icon in the **Service Payload** column, the **Transformed Payload** screen appears where you can view the details of services in the transformed payload.
Service Record Status	Displays the status of the service record.

By default, the **File Record Service Details** zone does not appear in the **File Record Details** screen. It appears when you click the **Broadcast** (📶) icon corresponding to the file record in the **Search Results** section of the **File Records** zone.

Related Topics

For more information on...	See...
File Record Details screen	File Record Details on page 1005
File Records zone.	File Records on page 1006

File Record Identifier Details

The **File Record Identifier Details** zone provides detailed information about the record identifiers associated with a specific file record. It contains the following columns:

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
Sequence Number	Displays the sequence number of the identifier that is configured in the **Transformation Detail** section of the **File Request Type**.
Field Name	Displays the field name of the identifier that is configured in the **Transformation Detail** section of the **File Request Type**.
Field Value	Displays the corresponding field value provided in the file record.

By default, the **File Record Identifier Details** zone does not appear in the **File Record Details** screen. It appears when you click the **Broadcast** ( ) icon corresponding to the file record in the **Search Results** section of the **File Records** zone.

Related Topics

For more information on...	See...
File Record Details screen	File Record Details on page 1005
File Records zone.	File Records on page 1006

File Record Status Transition

The **File Record Status Transition** zone provides detailed information about the status transitions of a file record throughout its life-cycle. It contains the following columns:

Column Name	Column Description
Status Transition Date Time	Displays the exact date and time when the file record status is transitioned. This provides an audit trail of status changes.
File Record Status	Displays the status of the record after the transition.
Status Update Reason	Displays the reason for which the status was updated.
Updated By	Displays the name of the user or system entity responsible for updating the status of the file record.

By default, the **File Record Status Transition** zone does not appear in the **File Record Details** screen. It appears when you click the **Broadcast** ( ) icon corresponding to the file record in the **Search Results** section of the **File Records** zone.

Related Topics

For more information on...	See...
File Record Details screen	File Record Details on page 1005
File Records zone.	File Records on page 1006

File Record Error Message History

The **File Record Error Message History** zone provides a comprehensive view of error details encountered during the processing of a file record. It contains the following columns:

Column Name	Column Description
Error Date Time	Displays the exact date and time when the error message occurred for the file record.
Error Message	Displays the detailed error message for the respective file record.

Copyright © 2024, Oracle and/or its affiliates.

By default, the **File Record Error Message History** zone does not appear in the **File Record Details** screen. It appears when you click the **Broadcast** (📶) icon corresponding to the file record in the **Search Results** section of the **File Records** zone.

Related Topics

For more information on...	See...
File Record Details screen	File Record Details on page 1005
File Records zone.	File Records on page 1006

Filtering a File Record

Procedure

To filter a file record:

1. Click the **Menu** link in the **Application** toolbar.
   A list appears.
2. From the **Main** menu, select **Integration** and then click **File Upload Dashboard**.
   The **File Upload Dashboard** screen appears.
3. Search for the file in the **Search File** zone whose records you want to view.
4. In the **Search Results** section, click the link in either of the record columns (such as, **Pending Records**, **Processed Records**, **Error Records**, **Retry Records**, **Ignore Records**, **Skipped Records**, **In Progress Records**, or **Records with Retry Limit Exceeded**) where the count is greater than zero corresponding to the file whose records you want to view.
   The **File Record Details** screen appears with the list of file records for the uploaded file.

   Note: You can filter the list using various search criteria (such as, **Record Identifier1**, **Record Identifier2**, **Record Identifier3**, **Record Identifier4**, **Record Identifier5**, **Operation**, **Batch Control**, **Batch Number**, **Record Status**, and **Ready for Archival**) available in the **Filter** area. By default, the **Filter** area is hidden. You can view the **Filter** area by clicking the **Filters** (▽) icon in the upper right corner of this zone or by clicking the **Expand Filters** button available in the **Search Results** section.
5. Enter the search criteria in the **File records** zone.
6. Click **Search**.
   The file record is filtered.

Related Topics

For more information on...	See...
File Upload Dashboard screen	File Upload Dashboard on page 997
Search File zone	Search File on page 998
File Record Details screen	File Record Details on page 1005
File Records zone	File Records on page 1006

Viewing the Details of a File Record

Procedure

To view the file record service details:

1. Search for the records in the **File Records** zone of the **File Record Details** screen.

Copyright © 2024, Oracle and/or its affiliates.

2. In the **Search Results** section, click the **Broadcast** (ᯤ) icon corresponding to the record whose details you want to view.
   The following zones appear:
   - File Record Service Details on page 1008
   - File Record Identifier Details on page 1008
   - File Record Status Transition on page 1009
   - File Record Error Message History on page 1009
3. View the details of the specific record in the respective zone.

Related Topics

For more information on...	See...
File Record Details screen	File Record Details on page 1005
File Records zone	File Records on page 1006

Updating the Status of a File Record

Procedure

To update the status of a file record:

1. Search for the records in the **File Records** zone of the **File Record Details** screen.
2. In the **Search Results** section, select the option corresponding to a record whose status you want to update.

   Note:

   At least one record should be selected to update the status of the file record.

   You can select all the records at once for updating by clicking the check-box available in the upper left corner of the table header in the **Search Results** section of the **File Records** zone.

3. Click **Update Record Status** button.
   The **File Record Update Reason** window appears.
4. Enter the reason for which you want to update the status of the record in the **Reason** field.

   Note: The status of a record which is in the **Error** or **Retry Limit Exceed** will be updated to **Retry**. Whereas, the status of a record which is in the **Pending** or **Retry** status will be updated to **Error**.

5. Click **OK**.
   The status of the record is updated.

Related Topics

For more information on...	See...
File Record Details screen	File Record Details on page 1005
File Records zone	File Records on page 1006

Marking a File Record for Archival

Procedure

To mark a file record for archival:

Copyright © 2024, Oracle and/or its affiliates.

1. Search for the records in the **File Records** zone of the **File Record Details** screen.
2. In the **Search Results** section, click the check-box corresponding to the record that to want to archive.

> **Note:**
>
> At least one record should be selected to update the status of the file record.
>
> You can select all the records at once for updating by clicking the check-box available in the upper left corner of the table header in the **Search Results** section of the **File Records** zone.

3. Click **Record Archival Update** option.
   The **File Upload Data Archival Tag Update** window appears.
4. Select **Y** from the **Mark Data Archival Flag** field to mark the file record for archival.

> **Note:** If you want to unmark the file record from archival, select the **N** option from the **Mark Data Archival Flag** field.

5. Click **OK**.
   The selected file is marked for archival.

Related Topics

For more information on...	See...
File Upload Dashboard screen	File Upload Dashboard on page 997
Search File zone	Search File on page 998
File Record Details screen	File Record Details on page 1005
File Records zone	File Records on page 1006

Viewing the Record Payload

Procedure

To view the record payload details:

1. Search for the record in the **File Records** zone of the **File Record Details** screen.
2. In the **Search Results** section, click the **View** (🔗) icon in the **Record Payload** column corresponding to the record whose record payload you want to view.
   The **Record Payload** window appears.
3. View the details of the record payload for the respective record in the **Record Payload** window.

Related Topics

For more information on...	See...
How to filter a file record	Filtering a File Record on page 1010
File Record Details screen	File Record Details on page 1005
File Records zone	File Records on page 1006

Viewing the Service Payload

Procedure

To view the service payload details:

Copyright © 2024, Oracle and/or its affiliates.

1. Search for the record in the **File Records** zone of the **File Record Details** screen.
2. In the **Search Results** section, click the **Broadcast** (📶) icon corresponding to the record whose service payload you want to view.
3. In the **File Record Service Details** zone, click the **View** (🔗) icon in the **Service Payload** column corresponding to the service whose details you want to view.
   The **Transformed Payload** window appears.
4. View the details of the service payload for the respective service in the **Transformed Payload** window.

Related Topics

For more information on...	See...
How to filter a file record	Filtering a File Record on page 1010
File Record Details screen	File Record Details on page 1005
File Records zone	File Records on page 1006

Viewing the Transformed Payload

Procedure

To view the transformed payload details:

1. Search for the record in the **File Records** zone of the **File Record Details** screen.
2. In the **Search Results** section, click the **View** (🔗) icon in the **Transformed Payload** column corresponding to the record whose transformed payload you want to view.
   The **Transformed Payload** window appears.
3. View the details of the transformed payload for the respective record in the **Transformed Payload** window.

Related Topics

For more information on...	See...
How to filter a file record	Filtering a File Record on page 1010
File Record Details screen	File Record Details on page 1005
File Records zone	File Records on page 1006

Copyright © 2024, Oracle and/or its affiliates.

Chapter 20

Extract Template

Topics:

- Extract Template

Oracle Revenue Management and Billing enables you to extract any data from the system to a file in the predefined format using the **Extract Template** feature. An extract template helps the system to determine the following:

- **Date Time Format** - The standard format in which the date and time should be extracted to a file.
- **File Formats** - You can extract the data in the following supported file formats:
  - Comma Separated Values (CSV)
  - Extensible Markup Language (XML)
  - Fixed Position (PRN)
  - JavaScript Object Notation (JSON)
  - Pipe Separated Values (PSV)
  - Tilde Separated Values (TSV)

  Here, you can specify the following:
  - Whether the root tag should be used in the XML or JSON file format
  - Whether the data should be enclosed within quotes in the CSV, PSV, or TSV file format
  - Whether the column header should displayed in the CSV, PSV, or TSV file format
  - Whether the source XML structure should be used in the XML or JSON file format
- **Services** - Here, you can specify a list of business objects or business services using which you want to extract the data from the system.
- **Extract Record List**

  - Here, you can specify the XPath of list for which sequence or positions should be reserved based on given record count. This is applicable for the CSV, PRN, PSV, and TSV file formats.
- **Field Mapping** - Here, you can specify the following information:
  - Field or node name that you want to appear in the output file
  - XPath of the source field from where you want to extract the data
  - XPath of the target field in which the data should be extracted (applicable for the XML and JSON file formats)
  - Start and end positions of a particular field (applicable for the PRN file format)
  - Whether you want to extract the data in this field
  - Whether you want to edit the extracted data using the supported functions
  - Whether you want to override the date and time format of a field

A new screen named **Extract Template** is introduced in this release. It enables you to define, edit, copy, and delete an extract template.

We have shipped one extract template named **INVOICE_EXTRACT_TEMP** which helps to extract the bill data in different formats. A new algorithm type named **C1-BLEXTTMPL** and algorithm named **C1-BLEXTTMPL** are shipped with product. If you want to extract the bill data using the **INVOICE_EXTRACT_TEMP** extract template, you need to accordingly set the **Extract Template Code** parameter in the **C1-BLEXTTMPL** algorithm and then attach the **C1-BLEXTTMPL** algorithm to the **Extract Algorithm** system event of the required bill route type.

Copyright © 2024, Oracle and/or its affiliates.

Extract Template

The **Extract Template** screen allows the you to view a list of extract templates which are already defined in the system. It also allows you to define, edit, copy, or delete an extract template. It contains the following zones:

- Extract Template List on page 1017
- Extract Template on page 1018

Related Topics

For more information on...	See...
How to define an extract template	Defining an Extract Template on page 1021
How to edit an extract template	Editing an Extract Template on page 1030
How to copy an extract template	Copying an Extract Template on page 1035
How to delete an extract template	Deleting an Extract Template on page 1039

Extract Template List

The **Extract Template List** zone lists the extract templates that are already defined in the system. It contains the following columns:

Column Name	Column Description
Extract Template	Displays the extract template.
Description	Displays the description of the extract template.
Status	Indicates the status of the extract template. The valid values are: • Active • Inactive **Note:** Extract templates which are in the **Inactive** status cannot be used for extracting information.
Edit	On clicking the **Edit** (✎) icon, the **Extract Template** screen appears where you can edit the details of the extract template.
Duplicate	On clicking the **Duplicate** (⧉) icon, the **Extract Template** screen appears where you can define a new extract template using an existing extract template.
Delete	On clicking the **Delete** (🗑) icon, you can delete the extract template.

Note: Pagination is used to display limited number of records in this zone. You can use the navigation links, such as **Previous** and **Next** to navigate between pages.

You can filter the list using various search criteria (such as, **Extract Template**, **Description** and **Status**) available in the **Filter** area. By default, the **Filter** area is hidden. You can view the **Filter** area by clicking the **Filters** (▽) icon in the upper right corner of this zone.

On clicking the **Broadcast** (📶) icon corresponding to an extract template, the **Extract Template** zone appears where you can view the details of the respective extract template.

Related Topics

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
How to edit an extract template	Editing an Extract Template on page 1030
How to copy an extract template	Copying an Extract Template on page 1035
How to delete an extract template	Deleting an Extract Template on page 1039
How to view the details of an extract template	Viewing the Details of an Extract Template on page 1039

Extract Template

The **Extract Template** zone displays the details of an extract template. It contains the following sections:

- **Main** - Displays the basic information about the extract template. It contains the following fields:

Field Name	Field Description
Extract Template	Displays the name of the extract template. **Note:** This is a read-only field.
Description	Displays the description of the extract template. **Note:** This is a read-only field.
File Header Required	Indicates whether header information is defined or not in the file for the extract template.
File Footer Required	Indicates whether footer information is defined or not in the file for the extract template.
Audit Log Required	Indicates whether transition of various statuses is logged in the ORMB system once each file request is processed.
Error Log Required	Indicates whether an error is logged in the ORMB system once each file request is processed.
File Archive Required	Indicates whether the file is relocated to another location once the processing takes place on the SFTP server.
File Encryption Required	Indicates whether to first decrypt and then extract the file on the SFTP server.
Status	Indicates the status of the extract template. The valid values are: • Active • Inactive **Note:** This is a read-only field.
Date/Time Format	Displays the format in which you want to define the date and time on extracting the information. **Note:** This is a read-only field.
Detailed Description	Displays additional information about the extract template. **Note:** This is a read-only field.

Copyright © 2024, Oracle and/or its affiliates.

- **Extract File Format** - Displays information about the file formats supported by the extract template for extracting information from the system. It contains the following columns:

Column Name	Column Description
File Format	Indicates the format in which you want to extract the information. The valid values are: • Comma Separated Values • Extensible Markup Language • Fixed Position • JavaScript Object Notation • Pipe Separated Values • Tilde Separated Values
File Extension	Displays the file extension selected corresponding to the selected file format.
Root Tag	Displays the name of the root tag in which the extracted information has to be enclosed. **Note:** This field is enabled when the **Extensible Markup Language** or **JavaScript Object Notation** option is selected from the **File Format** list.
Enclosed Quotes Required	Indicates whether you want to enclose the extracted information in double quotes. **Note:** This checkbox appears when the **Comma Separated Values**, **Pipe Separated Values** or **Tilde Separated Values** option is selected from the **File Format** list.
Column Header Required	Indicates whether you want the column headers in the extracted file or not. **Note:** This checkbox appears when the **Comma Separated Values**, **Pipe Separated Values** or **Tilde Separated Values** option is selected from the **File Format** list.
Do Not Extract	Indicates whether you want to exclude the corresponding file format from the extracted information.
Same as Source XML	Indicates whether the Target Field XPath is same as the Source Field XPath while mapping the fields. **Note:** This checkbox appears when the**Extensible Markup Language** or **JavaScript Object Notation** option is selected from the **File Format** list.

- **Services** - Displays the business services using which you want to extract information from the system. It contains the following columns:

Column Name	Column Description
Sequence Number	Displays the sequence number of the service.
Service Type	Indicates the service type you want to extract the information. The valid values are: • Business Service • Business Object
Schema Name	Displays the name of the service required to extract the information.

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
Dependent Service Name	Displays the name of the service which gives the details of the service hierarchy required for the information extract.
Post Processing Algorithm	Displays the name of the algorithm which is executed after the service is processed.

- **Extract Settings** - Displays the details about the record count of list element and fields mapped in the template. This section contains the following sub-sections:
  - **List Element Extract Details** - Displays the maximum list record count in a single data line in the extract information:

Column Name	Column Description
List Element Xpath	Displays the XPath of the list element for which the sequence / positions are reserved based on the given record count.
Maximum Records	Displays the number of records for a list element.

Note: This section appears when the **Comma Separated Values**, **Pipe Separated Values**, **Fixed Position** or **Tilde Separated Values** option is selected from the **File Format** list.

  - **Field Mapping** - Displays the details of the fields mapped in the extract template which are used to extract the information. It contains the following columns:

Column Name	Column Description
Sequence	Displays the sequence of the field.
Field Name	Displays the name of the field to be displayed in the extracted information file. **Note:** If the **Extensible Markup Language** option is selected from the **File Format** list then the node names gets displayed.
Source Field Path	Displays the XPath of the source field from where the information is to be extracted.
Target Field Path	Displays the Xpath of the target field into which the extracted information is to be saved. **Note:** This column is appears when the **Extensible Markup Language** or **JavaScript Object Notation** option is selected from the **File Format** list.
Start Position	Indicates the starting position of a particular field in the extracted file. **Note:** This column appears when the **Fixed Position** option is selected from the **File Format** list.
End Position	Indicates the ending position of a particular field in the extracted file. **Note:** This column appears when the **Fixed Position** option is selected from the **File Format** list.
Do Not Extract	Indicates whether you want to exclude corresponding field mapping from the extracted information.
Edit Input	Indicates whether you want to define the default value for the corresponding field.

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
	Note: If this checkbox is selected then the **Default Value** field is enabled.
Default Value	Indicates the default value defined for the corresponding field in the extract template.
	Note: This field is enabled when the **Edit Input** checkbox is selected.
Date Time Format	Specifies the format in which you want to define the date and time on extracting the information.

By default, the **Extract Template** zone does not appear in the **Extract Template** screen. It appears when you click the **Broadcast** (⦿) icon corresponding to a extract template in the **Extract Template List** zone.

Related Topics

For more information on...	See...
Extract Template screen	Extract Template on page 1017
Extract Template List zone	Extract Template List on page 1017
How to define the file formats for an extract template	Defining the File Formats for an Extract Template on page 1025
How to add business services or business objects in an extract template	Adding Business Services or Business Objects in an Extract Template on page 1027
How to define the record count for a list element	Defining the Record Count for a List Element on page 1028
How to map the fields in an extract template	Mapping Fields in an Extract Template on page 1029
How to view the details of an extract template	Viewing the Details of an Extract Template on page 1039

Defining an Extract Template

Prerequisites

To define an extract template you should have:

- Reference tables and columns defined in the application.
- Values defined for the **EXTRACT_TEMPLATE_SERVICE_TYPE** *a*nd **EXTRACT_TEMPLATE_FILE_FORMAT** lookup fields.
- An algorithm defined using the **C1-BLEXTTMPL** algorithm type.

Procedure

To define an extract template:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **E** and then click **Extract Template**.
   A sub-menu appears.
3. Click the **Add** option from the **Extract Template** sub-menu.

Copyright © 2024, Oracle and/or its affiliates.

The **Extract Template** screen appears. It contains the following section:

- **Main** - Used to specify the basic information about the extract template. It contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Extract Template	Used to specify the extract template.	Yes
Description	Used to specify the description of the extract template.	Yes
Status	Used to indicate the status of the extract template. The valid values are: • Active • Inactive	Yes
Date Time Format	Used to specify the standard date and time format you want the for the extract template.	Yes
Detailed Description	Used to specify the format in which you want to define the date and time in extracted file.	No

- **Extract Formats** - Used to specify the supported file formats and corresponding details required to extract the information. It contains the following fields:

Column Name	Column Description	Mandatory (Yes or No)
File Format	Used to indicate the file format selected to create an extract file. The valid values are: • Comma Separated Values • Extensible Markup Language • Fixed Position • JavaScript Object Notation • Pipe Separated Values • Tilde Separated Values	Yes
File Extension	Used to specify the file extension corresponding to the file format selected to extract information.	Yes
Root Tag	Used to specify the name of the root tag in which you want to enclose the extracted information. **Note:** This field is enabled only when you select **Extensible Markup Language** or **JavaScript Object Notation** option from the **File Format** list.	Yes (Conditional)
Enclosed Quotes Required	Used to indicate whether you want to enclose the extracted information in the double quotes. **Note:** This checkbox appears only when you select the **Comma Separated Values**, **Pipe Separated Values** or **Tilde Separated Values** option from the **File Format** list.	Yes (Conditional)
Column Header Required	Used to indicate whether you want the column headers for the extracted information.	Yes (Conditional)

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description	Mandatory (Yes or No)
	Note: • If this checkbox is selected then the field names of the mapped fields are used as the column headers for the extracted information. • This checkbox appears when you select the **Comma Separated Values**, **Pipe Separated Values** or **Tilde Separated Values** option from the **File Format** list.	
Same as Source XML	Used to indicate whether the target field XPath is same as the source field XPath while mapping the fields. **Note:** This checkbox appears when you select the **Comma Separated Values**, **Pipe Separated Values** or **Tilde Separated Values** option from the **File Format** list.	Yes (Conditional)

- **Services** - Used to specify the services used to extract information in the extract template from the system. It contains the following columns:

Column Name	Column Description	Mandatory (Yes or No)
Sequence Number	Used to specify the sequence number of the service.	Yes
Service Type	Used to indicate the service type you want to select to extract the information. The valid values are: • Business Service • Business Object	Yes
Service Name	Used to specify the name of the service you want to extract the information. **Note:** The **Search** (🔍) icon appears corresponding to this field. On clicking the **Search** icon, the **Schema Type/Schema Name Search** window appears.	Yes
Dependent Service Name	Used to specify the name of the service which gives the details of the service hierarchy required for the data extract.	No
Post Processing Algorithm	Used to specify the name of the algorithm corresponding to the selected service which will be executed post data extraction. **Note:** The **Search** (🔍) icon appears corresponding to this field. On clicking the **Search** icon, the **Algorithm Search** window appears and corresponding algorithm gets auto populated in the respective field.	No

- **List Element Extract Details** - Used to specify the XPath of the list elements and maximum record count of list *e*lements in a single data line in the extract data:

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description	Mandatory (Yes or No)
List Element XPath	Used to specify the XPath of the list element for which the sequence / positions are reserved based on the given record count.	No
Maximum Record	Used to specify the number of records for a list element.	No

Note: This section appears when you select the **Comma Separated Values**, **Pipe Separated Values**, **Fixed Position** or **Tilde Separated Values** option from the **File Format** list.

4. Enter the required details in the **Extract Formats** and **Services** sections.
5. Click **Generate**.
   Field mapping is generated with the details such as **Sequence**, **Field Name** and the **Source Field XPath** in the extract template based *o*n the selected the service.

   Field Mapping - Used to specify the details about the fields mapped in the extract template to extract the information. It contains the following columns:

Column Name	Column Description	Mandatory (Yes or No)
Sequence	Used to specify the display sequence of the field.	Yes
Field Name	Used to specify the name of the field to be displayed in the extracted data file. **Note:** If you select the **Extensible Markup Language** option from the **File Format** list then the node names gets displayed.	Yes
Start Position	Used to configure the starting position of a particular field. **Note:** This field appears when you select the **Fixed Position** option from the **File Format** list.	Yes
End Position	Used to configure the ending position of particular field. **Note:** This field appears when you select the **Fixed Position** option from the **File Format** list.	Yes
Source Field Path	Used to specify the XPath of the source field from where the data is to be extracted. **Note:** The **Search** (🔍) icon appears corresponding to this field. On clicking the **Search** icon, the **Field XPath Search** window appears and corresponding algorithm gets auto populated in the respective field.	Yes
Target Field Path	Used to specify the XPath of the target field into which the extracted data is to be saved.	Yes

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description	Mandatory (Yes or No)
	Note: This column appears when you select the **Extensible Markup Language** or **JavaScript Object Notation** option from the **File Format** list.	
Do not extract	Used to indicate whether you want to exclude corresponding field mapping from the extracted information.	Yes
Edit Input	Used to indicate whether you want to define the default value for the corresponding field. **Note:** If you select this checkbox then the **Default Value** field is enabled.	Yes
Default Value	Used to indicate the default value which you want to be displayed for the corresponding field in the extract template. **Note:** This field is enabled when you select the **Edit Input** checkbox.	Yes
Date Time Format	Used to specify the format in which you want to define the date and time in extracted file.	Yes

6. Click **Save**.
   The extract template is defined.

Tip: Alternatively, you can define an extract template by clicking the **Add** button in the **Page Title** area of the **Extract Template** screen.

Related Topics

For more information on...	See...
Extract Template screen	Extract Template on page 1017
How to define the file formats for an extract template	Defining the File Formats for an Extract Template on page 1025
How to add the business services or business objects in an extract template	Adding Business Services or Business Objects in an Extract Template on page 1027
How to define record count for a list element	Defining the Record Count for a List Element on page 1028
How to map the fields in an extract template	Mapping Fields in an Extract Template on page 1029

Defining the File Formats for an Extract Template

Prerequisites

To define the file formats for an extract template you should have:

- Values defined for the **EXTRACT_TEMPLATE_FILE_FORMAT**.

Procedure

To define the extract formats for an extract template:

Copyright © 2024, Oracle and/or its affiliates.

1. Ensure that the **Template Formats** section is expanded when you are defining, editing or copying an **Extract Template**.

   The **Template Formats** section contains the following fields in a grid:

Field Name	Field Description	Mandatory (Yes or No)
File Format	Used to indicate the format in which you want to extract the information. The valid values are: • Comma Separated Values • Extensible Markup Language • Fixed Position • JavaScript Object Notation • Pipe Separated Values • Tilde Separated Values	Yes
File Extension	Used to specify the file extension corresponding to the file format selected to extract information.	Yes
Root Tag	Used to specify the name of the root tag in which you want to enclose the extracted information. **Note:** This field is enabled only when you select **Extensible Markup Language** or **JavaScript Object Notation** option from the **File Format** list.	Yes (Conditional)
Enclosed Quotes Required	Used to indicate whether you want to enclose the extracted information in quotes. **Note:** The **Enclosed Quotes Required** field check box is enabled only when you select the **Comma Separated Values**, **Pipe Separated Values** or **Tilde Separated Values** option from the **File Format** list.	Yes (Conditional)
Column Header Required	Used to indicate whether you want the column headers for the extracted information. **Note:** • If this check box is selected then the field names of the mapped fields are used as the column headers for the extracted information. • The **Column Header Required** field check box is enabled only when you select the **Comma Separated Values**, **Pipe Separated Values** or **Tilde Separated Values** option from the **File Format** list.	Yes (Conditional)
Do Not Extract	Used to indicate whether you want to exclude the corresponding file format from the extracted information.	Yes
Same as Source XML	Used to indicate whether the target field XPath is same as the source field XPath while mapping the fields. **Note:** This checkbox appears when you select the **Comma Separated Values**, **Pipe Separated Values** or **Tilde Separated Values** option from the **File Format** list.	Yes (Conditional)

Copyright © 2024, Oracle and/or its affiliates.

2. Enter the required details in **Extract Formats** section.
3. If you want to define more than one extract format for the extract template, click the **Add** (+) icon and then *r*epeat step 2.

   Note: However, if you want to remove an extract format from the extract template, click the **Delete** (🗑) icon corresponding to the extract template.

4. Click **Save**.
   The extract format is defined for the extract template.

Related Topics

For more information on...	See...
How to define an extract template	Defining an Extract Template on page 1021
How to edit an extract template	Editing an Extract Template on page 1030
How to copy an extract template	Copying an Extract Template on page 1035

Adding Business Services or Business Objects in an Extract Template

Prerequisites

To add a business service or business object in an extract template, you should have:

- Reference tables and columns defined in the application.
- Values defined for the **EXTRACT_TEMPLATE_SERVICE_TYPE**.
- An algorithm defined using the **C1-BLEXTTMPL** algorithm type.

Procedure

To add a business service or business object in an extract template:

1. Ensure that the **Services** section is expanded when you are defining, editing, or copying an extract template.

   The **Services** section contains the following fields in a grid:

Column Name	Column Description	Mandatory (Yes or No)
Sequence Number	Used to specify the sequence number of the service.	Yes
Service Type	Used to indicate the service type you want to select to extract the information. The valid values are: • Business Service • Business Object	Yes
Service Name	Used to specify the name of the service you want to extract the information. **Note:** The **Search** (🔍) icon appears corresponding to this field. On clicking the **Search** icon, the **Schema Type/Schema Name Search** window appears.	Yes
Dependent Service Name	Used to specify the name of the service which gives the details of the service hierarchy required for the data extract.	No

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description	Mandatory (Yes or No)
Post Processing Algorithm	Used to specify the name of the algorithm corresponding to the selected service which will be executed post data extraction.	No
	Note: The **Search** (🔍) icon appears corresponding to this field. On clicking the **Search** icon, the **Algorithm Search** window appears and corresponding algorithm gets auto populated in the respective field.	

2. Enter the required details in the **Services** section.
3. If you want to define more than one service for the extract template, click the **Add** (+) icon and then *r*epeat step 2.

Note: However, if you want to remove a service from the extract template, click the **Delete** (🗑) icon corresponding to the service.

4. Click **Save**.
   The business service or business object is added in the extract template.

Related Topics

For more information on...	See...
How to define an extract template	Defining an Extract Template on page 1021
How to edit an extract template	Editing an Extract Template on page 1030
How to copy an extract template	Copying an Extract Template on page 1035

Defining the Record Count for a List Element

Procedure

To define the record count for a list element in an extract template:

1. Ensure that the **List Element Extract Details** section is expanded when you are defining, editing or copying *t*he **Extract Template** .

   The **List Element Extract Details** section contains the following fields:

Column Name	Column Description	Mandatory (Yes or No)
List Element XPath	Used to specify the XPath of the list element for which the sequence / positions are reserved based on the given record count.	No
Maximum Record	Used to specify the number of records for a list element.	No

Note: This section appears when you select the **Comma Separated Values**, **Pipe Separated Values**, **Fixed Position** or **Tilde Separated Values** option from the **File Format** list.

2. Enter the required details in the **List Element Extract Details** section.
3. If you want to define more than one service for the extract template, click the **Add** (+) icon and then *r*epeat step 2.

Note: However, if you want to remove a service from the extract template, click the **Delete** (🗑) icon corresponding to the service.

Copyright © 2024, Oracle and/or its affiliates.

4. Click **Save**.
   The record count is defined for a list element in an extract template.

Related Topics

For more information on...	See...
How to define an extract template	Defining an Extract Template on page 1021
How to edit an extract template	Editing an Extract Template on page 1030
How to copy an extract template	Copying an Extract Template on page 1035

Mapping Fields in an Extract Template

Prerequisites

To map fields in an extract template you should have:

- Business objects and business services defined in the application.
- Schema associated with the business object or the business service.

Procedure

To map fields in an extract template

1. Ensure that the **Field Mapping** section is expanded when you are defining, editing or copying an **Extract Template**.
2. Click **Generate**.

   Field mapping is generated with the details such as sequence, field names and the source field Xpath in the extract template based *o*n the selected the services.

   The **Field Mapping** section contains the following fields:

Column Name	Column Description	Mandatory (Yes or No)
Sequence	Used to specify the display sequence of the field.	Yes
Field Name	Used to specify the name of the field to be displayed in the extracted data file. **Note:** If you select the **Extensible Markup Language** option from the **File Format** list then the node names gets displayed.	Yes
Start Position	Used to configure the starting position of a particular field. **Note:** This field appears when you select the **Fixed Position** option from the **File Format** list.	Yes
End Position	Used to configure the ending position of particular field. **Note:** This field appears when you select the **Fixed Position** option from the **File Format** list.	Yes
Source Field Path	Used to specify the XPath of the source field from where the data is to be extracted.	Yes

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description	Mandatory (Yes or No)
	Note: The **Search** (🔍) icon appears corresponding to this field. On clicking the **Search** icon, the **Field XPath Search** window appears and corresponding algorithm gets auto populated in the respective field.	
Target Field Path	Used to specify the XPath of the target field into which the extracted data is to be saved. **Note:** This column appears when you select the **Extensible Markup Language** or **JavaScript Object Notation** option from the **File Format** list.	Yes
Do not extract	Used to indicate whether you want to exclude corresponding field mapping from the extracted information.	Yes
Edit Input	Used to indicate whether you want to define the default value for the corresponding field. **Note:** If you select this checkbox then the **Default Value** field is enabled.	Yes
Default Value	Used to indicate the default value which you want to be displayed for the corresponding field in the extract template. **Note:** This field is enabled when you select the **Edit Input** checkbox.	Yes
Date Time Format	Used to specify the format in which you want to define the date and time in exported file.	Yes

3. Enter the required details in the **Field Mapping** section of the extract template.
4. If you want to map more fields in the extract template, click the **Add** (+) icon and then repeat step 2.

   Note: However, if you want to remove a field from the extract template, click the **Delete** (🗑) icon corresponding to the service.

5. Click **Save**.
   The fields are mapped in the extract template.

Related Topics

For more information on...	See...
How to define an extract template	Defining an Extract Template on page 1021
How to edit an extract template	Editing an Extract Template on page 1030
How to copy an extract template	Copying an Extract Template on page 1035

Editing an Extract Template

Prerequisites

Copyright © 2024, Oracle and/or its affiliates.

To edit a extract template you should have:

- Fields and their corresponding default values are defined in the application.
- Reference tables and columns defined in the application.
- Values defined for the **EXTRACT_TEMPLATE_SERVICE_TYPE** *a*nd **EXTRACT_TEMPLATE_FILE_FORMAT**.
- An algorithm defined using the **C1-BLEXTTMPL** algorithm type.

Procedure

To edit an extract template:

1. Search for the extract template in the **Extract Template** screen.
2. In the **Search Results** section, click the **Edit** ( ) icon in the **Edit** column corresponding to the extract *t*emplate whose details you want to edit.
   The **Extract Template** window appears. It contains the following sections:
   - **Main** - Used to specify the basic information about the extract template. It contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Extract Template	Used to specify the extract template.	Yes
Description	Used to specify the description of the extract template.	Yes
Status	Used to indicate the status of the extract template. The valid values are: • Active • Inactive	Yes
Date Time Format	Used to specify the standard date and time format you want the for the extract template.	Yes
Detailed Description	Used to specify the format in which you want to define the date and time in extracted file.	No

   - **Extract Formats** - Used to specify the supported file formats and corresponding details required to extract the information. It contains the following fields:

Column Name	Column Description	Mandatory (Yes or No)
File Format	Used to indicate the file format selected to create an extract file. The valid values are: • Comma Separated Values • Extensible Markup Language • Fixed Position • JavaScript Object Notation • Pipe Separated Values • Tilde Separated Values	Yes
File Extension	Used to specify the file extension corresponding to the file format selected to extract information.	Yes
Root Tag	Used to specify the name of the root tag in which you want to enclose the extracted information.	Yes (Conditional)

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description	Mandatory (Yes or No)
	Note: This field is enabled only when you select **Extensible Markup Language** or **JavaScript Object Notation** option from the **File Format** list.	
Enclosed Quotes Required	Used to indicate whether you want to enclose the extracted information in the double quotes. **Note:** This checkbox appears only when you select the **Comma Separated Values**, **Pipe Separated Values** or **Tilde Separated Values** option from the **File Format** list.	Yes (Conditional)
Column Header Required	Used to indicate whether you want the column headers for the extracted information. **Note:** • If this checkbox is selected then the field names of the mapped fields are used as the column headers for the extracted information. • This checkbox appears when you select the **Comma Separated Values**, **Pipe Separated Values** or **Tilde Separated Values** option from the **File Format** list.	Yes (Conditional)
Same as Source XML	Used to indicate whether the target field XPath is same as the source field XPath while mapping the fields. **Note:** This checkbox appears when you select the **Comma Separated Values**, **Pipe Separated Values** or **Tilde Separated Values** option from the **File Format** list.	Yes (Conditional)

- **Services** - Used to specify the services used to extract the information in the extract template. It contains the following columns:

Column Name	Column Description	Mandatory (Yes or No)
Sequence Number	Used to specify the sequence number of the service.	Yes
Service Type	Used to indicate the service type you want to select to extract the information. The valid values are: • Business Service • Business Object	Yes
Service Name	Used to specify the name of the service you want to extract the information. **Note:** The **Search** (🔍) icon appears corresponding to this field. On clicking the **Search** icon, the **Schema Type/Schema Name Search** window appears.	Yes

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description	Mandatory (Yes or No)
Dependent Service Name	Used to specify the name of the service which gives the details of the service hierarchy required for the data extract.	No
Post Processing Algorithm	Used to specify the name of the algorithm corresponding to the selected service which will be executed post data extraction. **Note:** The **Search** (🔍) icon appears corresponding to this field. On clicking the **Search** icon, the **Algorithm Search** window appears and corresponding algorithm gets auto populated in the respective field.	No

- **List Element Extract Details** - Used to specify the XPath of the list elements and maximum record count of list *e*lements in a single data line in the extract data:

Column Name	Column Description	Mandatory (Yes or No)
List Element XPath	Used to specify the XPath of the list element for which the sequence / positions are reserved based on the given record count.	No
Maximum Record	Used to specify the number of records for a list element.	No

Note: This section appears when you select the **Comma Separated Values**, **Pipe Separated Values**, **Fixed Position** or **Tilde Separated Values** option from the **File Format** list.

3. Enter the required details in the **Extract Formats** and **Services** sections.
4. Click **Generate**.
   Field mapping is generated with the details such as **Sequence**, **Field Name** and the **Source Field XPath** in the extract template based *o*n the selected the service.

   Field Mapping - Used to specify the details about the fields mapped in the extract template to extract the information. It contains the following columns:

Column Name	Column Description	Mandatory (Yes or No)
Sequence	Used to specify the display sequence of the field.	Yes
Field Name	Used to specify the name of the field to be displayed in the extracted data file. **Note:** If you select the **Extensible Markup Language** option from the **File Format** list then the node names gets displayed.	Yes
Start Position	Used to configure the starting position of a particular field. **Note:** This field appears when you select the **Fixed Position** option from the **File Format** list.	Yes
End Position	Used to configure the ending position of particular field.	Yes

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description	Mandatory (Yes or No)
	Note: This field appears when you select the **Fixed Position** option from the **File Format** list.	
Source Field Path	Used to specify the XPath of the source field from where the data is to be extracted. **Note:** The **Search** (🔍) icon appears corresponding to this field. On clicking the **Search** icon, the **Field XPath Search** window appears and corresponding algorithm gets auto populated in the respective field.	Yes
Target Field Path	Used to specify the XPath of the target field into which the extracted data is to be saved. **Note:** This column appears when you select the **Extensible Markup Language** or **JavaScript Object Notation** option from the **File Format** list.	Yes
Do not extract	Used to indicate whether you want to exclude corresponding field mapping from the extracted information.	Yes
Edit Input	Used to indicate whether you want to define the default value for the corresponding field. **Note:** If you select this checkbox then the **Default Value** field is enabled.	Yes
Default Value	Used to indicate the default value which you want to be displayed for the corresponding field in the extract template. **Note:** This field is enabled when you select the **Edit Input** checkbox.	Yes
Date Time Format	Used to specify the format in which you want to define the date and time in extracted file.	Yes

5. Modify the required details in all sections.
6. Click **Save**.
   The changes made to the extract template are saved.

For more information on...	See...
Extract Template screen	Extract Template on page 1017
Extract Template List zone	Extract Template List on page 1017
How to define the file formats for an extract template	Defining the File Formats for an Extract Template on page 1025
How to add the business services or business objects in an extract template	Adding Business Services or Business Objects in an Extract Template on page 1027
How to define record count for a list element	Defining the Record Count for a List Element on page 1028

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
How to map the fields in an extract template	Mapping Fields in an Extract Template on page 1029

Copying an Extract Template

Instead of creating an extract template from scratch, you can create a new extract template using an existing extract template. This is possible through copying an extract template. On copying an extract template, the details including the fields are copied to the new extract template. You can then edit the details, if required.

Prerequisites

To copy an extract template you should have:

- An extract template defined in the system.
- Fields and their corresponding default values are defined in the application.
- Reference tables and columns defined in the application.
- Values defined for the **EXTRACT_TEMPLATE_SERVICE_TYPE** *a*nd **EXTRACT_TEMPLATE_FILE_FORMAT**.
- An algorithm defined using the **C1-BLEXTTMPL** algorithm type.

Procedure

To copy an extract template:

1. Search for the extract template in the **Extract Template** screen.
2. In the **Search Results** section, click the **Duplicate** (⧉) icon in the **Duplicate** column corresponding to *t*he extract template whose copy you want to create

   The **Extract Template** screen appears. It contains the following sections:

   - **Main** - Used to specify the basic information about the extract template. It contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Extract Template	Used to specify the extract template.	Yes
Description	Used to specify the description of the extract template.	Yes
Status	Used to indicate the status of the extract template. The valid values are: • Active • Inactive	Yes
Date Time Format	Used to specify the standard date and time format you want the for the extract template.	Yes
Detailed Description	Used to specify the format in which you want to define the date and time in extracted file.	No

   - **Extract Formats** - Used to specify the supported file formats and corresponding details required to extract the information. It contains the following fields:

Column Name	Column Description	Mandatory (Yes or No)
File Format	Used to indicate the file format selected to create an extract file. The valid values are: • Comma Separated Values • Extensible Markup Language	Yes

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description	Mandatory (Yes or No)
	• Fixed Position • JavaScript Object Notation • Pipe Separated Values • Tilde Separated Values	
File Extension	Used to specify the file extension corresponding to the file format selected to extract information.	Yes
Root Tag	Used to specify the name of the root tag in which you want to enclose the extracted information. **Note:** This field is enabled only when you select **Extensible Markup Language** or **JavaScript Object Notation** option from the **File Format** list.	Yes (Conditional)
Enclosed Quotes Required	Used to indicate whether you want to enclose the extracted information in the double quotes. **Note:** This checkbox appears only when you select the **Comma Separated Values**, **Pipe Separated Values** or **Tilde Separated Values** option from the **File Format** list.	Yes (Conditional)
Column Header Required	Used to indicate whether you want the column headers for the extracted information. **Note:** • If this checkbox is selected then the field names of the mapped fields are used as the column headers for the extracted information. • This checkbox appears when you select the **Comma Separated Values**, **Pipe Separated Values** or **Tilde Separated Values** option from the **File Format** list.	Yes (Conditional)
Same as Source XML	Used to indicate whether the target field XPath is same as the source field XPath while mapping the fields. **Note:** This checkbox appears when you select the **Comma Separated Values**, **Pipe Separated Values** or **Tilde Separated Values** option from the **File Format** list.	Yes (Conditional)

- **Services** - Used to specify the services used to extract information in the extract template from the system. It contains the following columns:

Column Name	Column Description	Mandatory (Yes or No)
Sequence Number	Used to specify the sequence number of the service.	Yes
Service Type	Used to indicate the service type you want to select to extract the information. The valid values are: • Business Service	Yes

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description	Mandatory (Yes or No)
	• Business Object	
Service Name	Used to specify the name of the service you want to extract the information. **Note:** The **Search** (🔍) icon appears corresponding to this field. On clicking the **Search** icon, the **Schema Type/Schema Name Search** window appears.	Yes
Dependent Service Name	Used to specify the name of the service which gives the details of the service hierarchy required for the data extract.	No
Post Processing Algorithm	Used to specify the name of the algorithm corresponding to the selected service which will be executed post data extraction. **Note:** The **Search** (🔍) icon appears corresponding to this field. On clicking the **Search** icon, the **Algorithm Search** window appears and corresponding algorithm gets auto populated in the respective field.	No

- **List Element Extract Details** - Used to specify the XPath of the list elements and maximum record count of list elements in a single data line in the extract data:

Column Name	Column Description	Mandatory (Yes or No)
List Element XPath	Used to specify the XPath of the list element for which the sequence / positions are reserved based on the given record count.	No
Maximum Record	Used to specify the number of records for a list element.	No

Note: This section appears when you select the **Comma Separated Values**, **Pipe Separated Values**, **Fixed Position** or **Tilde Separated Values** option from the **File Format** list.

3. Enter the required details in the **Extract Formats** and **Services** sections.
4. Click **Generate**.
   Field mapping is generated with the details such as **Sequence**, **Field Name** and the **Source Field XPath** in the extract template based on the selected the service.

   Field Mapping - Used to specify the details about the fields mapped in the extract template to extract the information. It contains the following columns:

Column Name	Column Description	Mandatory (Yes or No)
Sequence	Used to specify the display sequence of the field.	Yes
Field Name	Used to specify the name of the field to be displayed in the extracted data file. **Note:** If you select the **Extensible Markup Language** option from the **File Format** list then the node names gets displayed.	Yes

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description	Mandatory (Yes or No)
Start Position	Used to configure the starting position of a particular field. **Note:** This field appears when you select the **Fixed Position** option from the **File Format** list.	Yes
End Position	Used to configure the ending position of particular field. **Note:** This field appears when you select the **Fixed Position** option from the **File Format** list.	Yes
Source Field Path	Used to specify the XPath of the source field from where the data is to be extracted. **Note:** The **Search** (🔍) icon appears corresponding to this field. On clicking the **Search** icon, the **Field XPath Search** window appears and corresponding algorithm gets auto populated in the respective field.	Yes
Target Field Path	Used to specify the XPath of the target field into which the extracted data is to be saved. **Note:** This column appears when you select the **Extensible Markup Language** or **JavaScript Object Notation** option from the **File Format** list.	Yes
Do not extract	Used to indicate whether you want to exclude corresponding field mapping from the extracted information.	Yes
Edit Input	Used to indicate whether you want to define the default value for the corresponding field. **Note:** If you select this checkbox then the **Default Value** field is enabled.	Yes
Default Value	Used to indicate the default value which you want to be displayed for the corresponding field in the extract template. **Note:** This field is enabled when you select the **Edit Input** checkbox.	Yes
Date Time Format	Used to specify the format in which you want to define the date and time in extracted file.	Yes

5. Define, edit, or remove details from the extract template if required.
6. Click **Save**.
   The new extract template is defined.

For more information on...	See...
Extract Template screen	Extract Template on page 1017

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
Extract Template List zone	Extract Template List on page 1017
How to define the file formats for an extract template	Defining the File Formats for an Extract Template on page 1025
How to add the business services or business objects in an extract template	Adding Business Services or Business Objects in an Extract Template on page 1027
How to define record count for a list element	Defining the Record Count for a List Element on page 1028
How to map the fields in an extract template	Mapping Fields in an Extract Template on page 1029

Deleting an Extract Template

Procedure

To delete an extract template:

1. Search for the extract template in the **Extract Template** screen.
2. In the **Search Results** section, click the **Delete** (🗑) icon in the **Delete** column corresponding to the *e*xtract template that you want to delete.
   A message appears confirming whether you want to delete the extract template.
3. Click **OK**.
   The extract template is deleted.

Related Topics

For more information on...	See...
Extract Template screen	Extract Template on page 1015
Extract Template List zone	Extract Template List on page 1017

Viewing the Details of an Extract Template

Procedure

To view the details of an extract template

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **E** and then click **Extract Template**.
   A sub-menu appears.
3. Click the **Search** option from the **Extract Template** sub-menu.
   The **Extract Template** screen appears.
4. In the **Extract Template List** zone, click the **Broadcast** (📶) icon corresponding to the extract template *w*hose details you want to view.
   The **Extract Template** zone appears.
5. View the details of the extract template in the **Extract Template** zone.

Related Topics

For more information on...	See...
Extract Template screen	Extract Template on page 1017

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
Extract Template List zone	Extract Template List on page 1017
Extract Template zone	Extract Template on page 1018

Chapter 21

Batch Scheduler

Topics:

- Prerequisites
- Oracle Scheduler Interface Architecture
- DBMS Import/Export
- Scheduler Chain
- Scheduler Job
- Scheduler Job Monitor
- Scheduler Job Monitor Dashboard
- Scheduler Job Details
- Scheduler Program
- Scheduler Schedule

The Batch Scheduler is a set of processes and objects that are defined and executed within the ORMB framework using the objects implemented by the **DBMS_SCHEDULER** package that is embedded in every installation of the Oracle Database. ORMB uses the DBMS scheduler objects to schedule Jobs/ Batches in the background.

Some features of ORMB DBMS scheduler include:

- User Interface to help define workflow and/or job dependencies
- User Interface to submit/schedule, monitor and administration of Batch Jobs

The DBMS Scheduler supports the following scheduling methods:

- Time based scheduling - Scheduling job based upon dates and times
- Dependency scheduling - Scheduling based upon job dependencies using Chains

The ORMB DBMS scheduler uses the following objects:

- **Program**: The lowest object in a scheduler is the Program. A program describes what is to be run by the schedule. The program object includes a definition of the physical object as well as arguments to execute them.
- **Job**: An instance of a program, is a Job. The job is a collection of metadata that defines the program to execute as well as where to execute it (for remote executions), its related schedule (which dictates when it is executed) and any related information. Jobs are subject to scheduling using the time based, event based and/or dependency of other jobs.
- **Schedule**: The Schedule object defines when and how many times the job is executed. The schedule object uses a rich calendaring syntax to define repeating schedules. Jobs also contain data used for prioritizing as well as resource profiles including support for Windows and Resource Manager.
- **Chains**: Chains represent the sequences of jobs expressing the dependencies. Chains define steps which link a job or file watcher and also contains Rules to decide the sequence and outcomes based upon the state of another job in the chain. For example, Job B needs to run only if Job A has completed successfully. Chains can support multiple branches and also chains can include other chains for greater reuse.

Advantages of using Oracle Scheduler Interface:

- License free implementation
- ORMB embedded user interfaces for setting up Batch Scheduling
- Less Administration
- Supports High Availability
- Maximum Flexibility

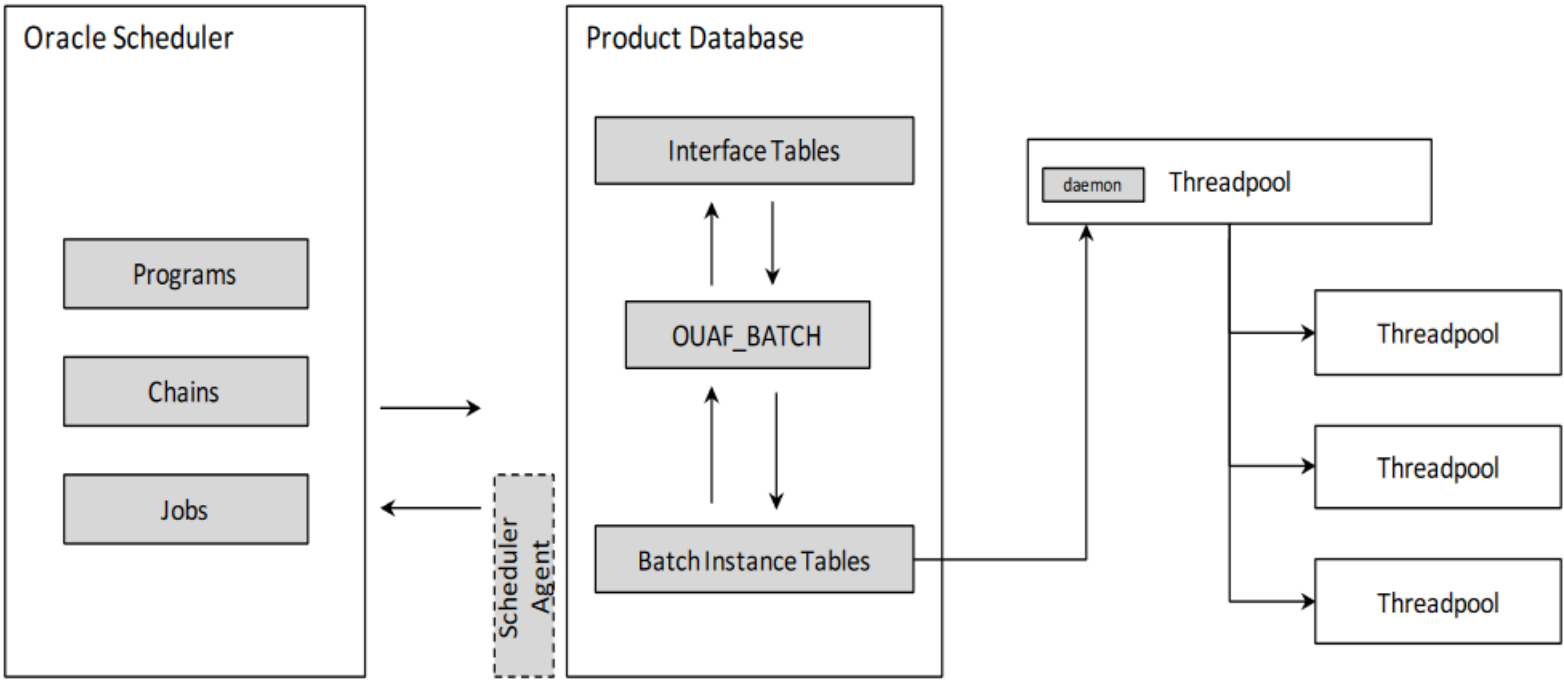
Prerequisites

To setup the batch scheduler feature, you need to do following:

- Scheduler program defined in the application
- Scheduler job defined in the application
- Scheduler schedule defined in the application
- Scheduler chain defined in the application

Oracle Scheduler Interface Architecture

The Oracle Scheduler interface has a flexible architecture that exploits the underlying features of the product to execute and monitor background processes.



From an architecture perspective, the following applies:

- The Oracle Scheduler objects are held as dictionary objects in the Schedule Administrator schema. At a minimum, Oracle Scheduler uses Programs, Jobs and Chains. Other Schedule objects can be defined as necessary.
- The Oracle Scheduler interface is installed on the product schema which includes a number of interface tables, the OUAF_BATCH pl/sql package and permissions to product batch tables.

DBMS Import/Export

The **DBMS Import Export** screen allows you to export DBMS scheduler data from one environment and import it to a new environment. It also allows you to decide whether records should be overwritten if already exists in the environment or not.

Related Topics

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
How to export the DBMS scheduler data	Exporting the DBMS Scheduler Data on page 1043
How to import the DBMS scheduler data	Importing the DBMS Scheduler Data on page 1044

Exporting the DBMS Scheduler Data

Prerequisites

To export DBMS scheduler data, you should have:

- DBMS scheduler files and folder.

Procedure

To export DBMS scheduler data:

1. Click the **Menu** link in the **Application** toolbar.
   A list appears.
2. From the **Main** menu, select**Batch Scheduler** and then click the **DBMS Import/Export** sub-menu.
   The **Import/ Export DBMS Data** screen appears.
3. Select **Export** option. It contains following fields:

Field Name	Field Description	Mandatory (Yes or No)
File Path	Used to specify the path of the file whose data you want to export from the environment. The path always starts with either: • @SHARED_DIR: The configured path of shared directory • @INSTALL_DIR: The configured path of installation directory, defined with the property: spl.runtime.environ.SPLEBASE in spl.properties file	Yes
Compress Files	Used to specify whether you want to compress all the files in the folder or not. **Note:** If the check box is selected then one zipped folder having all files is created.	No

4. Enter the file path in the **File Path** field which you want to export from the environment.
5. Select **Compress Files** check box to compress the files.
6. Click **Export**.
   A folder having all files is exported.

Related Topics

For more information on...	See...
DBMS Import/Export	DBMS Import/Export on page 1042

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
How to import the DBMS scheduler data	Importing the DBMS Scheduler Data on page 1044

Importing the DBMS Scheduler Data

Prerequisites

To import DBMS scheduler data, you should have:

- DBMS scheduler files and folder.

Procedure

To import DBMS scheduler data:

1. Click the **Menu** link in the **Application** toolbar.
   A list appears.
2. From the **Main** menu, select**Batch Scheduler** and then click the **DBMS Import/Export** sub-menu.
   The **Import/ Export DBMS Data** screen appears.
3. Select **Import** option. It contains following fields:

Field Name	Field Description	Mandatory (Yes or No)
File Path	Used to specify the path of the file whose data you want to import in the environment. The path always starts with either: • @SHARED_DIR: The configured path of shared directory • @INSTALL_DIR: The configured path of installation directory, defined with the property: spl.runtime.environ.SPLEBASE in spl.properties file	Yes
Overwrite Existing Records	Used to specify whether you want to overwrite the existing data while importing or skip the records which is already exists.. **Note:** By default, system skips the records which are already exists.	No

4. Enter the file path in the **File Path** field which you want to import in the environment.
5. Select **Overwrite Existing Records** check box to overwrite the existing data.
6. Click **Import**.
   A folder having all files is imported.

Related Topics

For more information on...	See...
DBMS Import/Export	DBMS Import/Export on page 1042
How to import the DBMS scheduler data	Importing the DBMS Scheduler Data on page 1044

Copyright © 2024, Oracle and/or its affiliates.

Scheduler Chain

The **Scheduler Chain** screen allows you to search a scheduler chain using various search criteria. It also allows you to define, view, edit, and delete a scheduler chain. It contains the following zone:

- Search Scheduler Chain on page 1045

Related Topics

For more information on...	See...
How to search for a scheduler chain	Searching for a Scheduler Chain on page 1046
How to define a scheduler chain	Defining a Scheduler Chain on page 1046
How to view the scheduler chain details	Viewing a Scheduler Chain Details on page 1048
How to edit a scheduler chain	Editing a Scheduler Chain on page 1049
How to delete a scheduler chain	Deleting a Scheduler Chain on page 1050

Search Scheduler Chain

The **Search Scheduler Chain** zone allows you to search for a scheduler chain using various search criteria. It contains the following sections:

- **Search Criteria** - The **Search Criteria** contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Chain Name	Used to search for scheduler chain having a particular name.	No
Detailed Description	Used to search for scheduler chain having the particular description.	No

- **Search Results** - On clicking the **Search** button, the search results appear based on the specified search criteria. The **Search Results** section contains the following columns:

Column Name	Column Description
Chain Name	Displays the name of the scheduler chain.
	Note: It has link. On clicking the link, the **Scheduler Chain** screen appears where you can view the details of the respective scheduler chain.
Detailed Description	Displays the detail description of the scheduler chain.
Edit	On clicking the **Edit** ( ) icon, the **Scheduler Chain** screen appears where you can edit the details of the existing scheduler chain.
Delete	On clicking the **Delete** ( ) icon, you can delete the existing scheduler chain.

Related Topics

For more information on...	See...
How to search for a scheduler chain	Searching for a Scheduler Chain on page 1046
How to define a scheduler chain	Defining a Scheduler Chain on page 1046

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
How to view the scheduler chain details	Viewing a Scheduler Chain Details on page 1048
How to edit a scheduler chain	Editing a Scheduler Chain on page 1049
How to delete a scheduler chain	Deleting a Scheduler Chain on page 1050

Searching for a Scheduler Chain

Prerequisites

To search a scheduler chain, you should have:

- Scheduler chain defined in the application.

Procedure

To search a scheduler chain:

1. Click the **Menu** link in the **Application** toolbar.
   A list appears.
2. From the **Main** menu, select**Batch Scheduler** and then click the **Scheduler Chain** sub-menu.
   The **Scheduler Chain** screen appears.
3. Enter the search criteria in the **Search Scheduler Chain** zone.

   Note: ORMB search engine supports wildcard search, where you can substitute the percentage (%) symbol as a stand in for any word or letter in a search criteria. You can use the '%' wildcard character in all input fields except the date and ID fields. The '%' wildcard character is suffixed automatically at the end of the partial search criteria. Therefore, you may or may not specify the wildcard character at the end of the partial search criteria. However, you have to prefix the wildcard character manually wherever required.
4. Click **Search** button.
   A list of scheduler chain that meet the search criteria appears in the **Search Results** section.

Related Topics

For more information on...	See...
Search Scheduler Chain zone	Search Scheduler Chain on page 1045
How to define a scheduler chain	Defining a Scheduler Chain on page 1046
How to view the scheduler chain details	Viewing a Scheduler Chain Details on page 1048
How to edit a scheduler chain	Editing a Scheduler Chain on page 1049
How to delete a scheduler chain	Deleting a Scheduler Chain on page 1050

Defining a Scheduler Chain

Prerequisites

To add a scheduler chain, you should have:

- Program name defined in the system.

Procedure

To add a scheduler chain:

1. Click the **Menu** link in the **Application** toolbar.

Copyright © 2024, Oracle and/or its affiliates.

A list appears.

2. From the **Main** menu, select**Batch Scheduler** and then click the **Scheduler Chain** sub-menu.
   The **Scheduler Chain** screen appears.
3. Click **Add** link in the page title area of the **Search Scheduler Chain** screen.
   The **Scheduler Chain** screen appears. It contains the following fields:

j

Field Name	**Field Description**	**Mandatory (Yes or No)**
Chain Name	Used to specify the name of scheduler chain.	Yes
	Note: You should prefix the chain name with **CM** (customer modification) while defining the chain on the deployed environment. For example: The changes performed during and after the installation of the product. You should prefix the chain name with **C1** while defining the chain on the development environment. For example: The changes performed on customer modification data.	
	The valid formats are: • C1_BILL • C1BILL • CM_BILL • CMBILL The invalid formats are: • C1-BILL • C1 BILL • CM-BILL • CM BILL	
Detailed Description	Used to specify the description of the scheduler chain.	No
Step Name	Used to specify the name of the step which you want to add in the particular chain.	Yes
	Note: The step refer to the particular program, which executes for that step. A rule is attached to each step to identify the dependency to other step and to the condition for when the step should be executed.	
	For Example: C1-PNDBL, C1-BLGE, and C1-BLPPR are the programs which should be executed in a specific	

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	sequence. The second program should be executed only when first program is completed successfully. Below are the steps to define the condition: • The condition of the first step should always true. So the scheduler understands that this step need to execute first before second step start for execution. • The condition for first program is mentioned as "STEP1" SUCCEEDED • In case, if you want to execute the second program even if the first program is failed to execute. The condition is mentioned as "STEP1" SUCCEEDED OR "STEP1" FAILED • The step names could be anything only the name of the last step should be mentioned as END.	
Program Name	Used to indicate the name of scheduler program. **Note:** The **Search** (🔍) icon appears corresponding to this field. On clicking the **Search** icon, the **Scheduler Program Search** window appears.	Yes
Execution Condition	Used to specify the condition for the particular step.	Yes

4. Click **Add** (icon corresponding to the respective step, to add the next step.
5. Click **Save**.
   The scheduler chain is added.

Related Topics

For more information on...	See...
Search Scheduler Chain zone	Search Scheduler Chain on page 1045
How to search for a scheduler chain	Searching for a Scheduler Chain on page 1046
How to view the scheduler chain details	Viewing a Scheduler Chain Details on page 1048
How to edit a scheduler chain	Editing a Scheduler Chain on page 1049
How to delete a scheduler chain	Deleting a Scheduler Chain on page 1050

Viewing a Scheduler Chain Details

Procedure

To view a scheduler chain:

1. Search for the scheduler chain in the **Search Scheduler Chain** zone of the **Scheduler Chain** screen.

Copyright © 2024, Oracle and/or its affiliates.

> **Note:** ORMB search engine supports wildcard search, where you can substitute the percentage (%) symbol as a stand in for any word or letter in a search criteria. You can use the '%' wildcard character in all input fields except the date and ID fields. The '%' wildcard character is suffixed automatically at the end of the partial search criteria. Therefore, you may or may not specify the wildcard character at the end of the partial search criteria. However, you have to prefix the wildcard character manually wherever required.

2. In the **Search Result** section, click the link in the **Chain Name** column whose details you want to view.
   The **Scheduler Chain** screen appears.
3. View the details of scheduler chain in the **Scheduler Chain** screen.

Related Topics

For more information on...	See...
Search Scheduler Chain zone	Search Scheduler Chain on page 1045
How to search for a scheduler chain	Searching for a Scheduler Chain on page 1046
How to define a scheduler chain	Defining a Scheduler Chain on page 1046
How to edit a scheduler chain	Editing a Scheduler Chain on page 1049
How to delete a scheduler chain	Deleting a Scheduler Chain on page 1050

Editing a Scheduler Chain

Procedure

To edit a scheduler chain:

1. Search for the scheduler chain in the **Scheduler Chain** screen whose details you want to edit.
2. In the **Search Results** section, click the **Edit** (✎) icon in the **Edit** column corresponding to the scheduler chain whose details you want to edit.
   The **Scheduler Chain** screen appears. It contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Chain Name	Displays the name of scheduler chain.	No
Detailed Description	Used to specify the description of the scheduler chain.	No
Step Name	Used to specify the name of the step which you want to edit for that particular chain. **Note:** The step refer to the program, which executes for that step. A rule is attached to each step to identify the dependency to other step and to the condition for when the step should be executed. For Example: C1-PNDBL, C1-BLGE, and C1-BLPPR are the programs which should be executed in a specific sequence. The second program should be executed only when first program is completed	Yes

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	successfully. Below are the steps to define the condition: • The condition of the first step should always true. So that the scheduler understands that this step need to execute first before second step start for execution. • The condition for first program is mentioned as "STEP1" SUCCEEDED • In case, if you want to execute the second program even if the first program is failed to execute. The condition is mentioned as "STEP1" SUCCEEDED OR "STEP1" FAILED • The step names could be anything only the name of the last step should be mentioned as END.	
Program Name	Used to indicate the name of scheduler program. **Note:** The **Search** (🔍) icon appears corresponding to this field. On clicking the **Search** icon, the **Scheduler Program Search** window appears.	Yes
Execution Condition	Used to specify the condition for the particular step.	Yes

3. Click **Add** (icon corresponding to the respective step, to add the next step.
4. Click **Delete** (🗑) icon corresponding to the respective step, to delete the step.
5. Click **Save**.
   The scheduler chain is modified.

Related Topics

For more information on...	See...
Search Scheduler Chain zone	Search Scheduler Chain on page 1045
How to search for a scheduler chain	Searching for a Scheduler Chain on page 1046
How to define a scheduler chain	Defining a Scheduler Chain on page 1046
How to view the scheduler chain details	Viewing a Scheduler Chain Details on page 1048
How to delete the scheduler chain	Deleting a Scheduler Chain on page 1050

Deleting a Scheduler Chain

Procedure

To delete a scheduler chain:

1. Search for the scheduler chain in the **Search Scheduler Chain** zone of the **Scheduler Chain** screen.

Copyright © 2024, Oracle and/or its affiliates.

> **Note:** ORMB search engine supports wildcard search, where you can substitute the percentage (%) symbol as a stand in for any word or letter in a search criteria. You can use the '%' wildcard character in all input fields except the date and ID fields. The '%' wildcard character is suffixed automatically at the end of the partial search criteria. Therefore, you may or may not specify the wildcard character at the end of the partial search criteria. However, you have to prefix the wildcard character manually wherever required.

2. In the **Search Result** section, click the **Delete** (🗑) icon in the **Delete** column corresponding to the scheduler chain which you want to delete.
   A message appears confirming whether you want to delete the scheduler chain.
3. Click **Ok**.
   The scheduler chain is deleted.

Related Topics

For more information on...	See...
Search Scheduler Chain zone	
How to search for a scheduler chain	
How to define a scheduler chain	
How to view the scheduler chain details	
How to edit a scheduler chain	

Scheduler Job

The **Scheduler Job** screen allows you to search a scheduler job using various search criteria. It also allows you to define, view, edit, delete and submit a scheduler job. It contains the following zone:

- Search Scheduler Job on page 1051

Related Topics

For more information on...	See...
How to search for a scheduler job	Searching for a Scheduler Job on page 1052
How to define a scheduler job	Defining a Scheduler Job on page 1053
How to view the scheduler job details	Viewing a Scheduler Job Details on page 1055
How to edit a scheduler job	Editing a Scheduler Job on page 1055
How to delete a scheduler job	Deleting a Scheduler Job on page 1056
How to submit a scheduler job	Submitting a Scheduler Job on page 1057

Search Scheduler Job

The **Search Scheduler Job** zone allows you to search for a scheduler job using various search criteria. It contains the following sections:

- **Search Criteria** - The **Search Criteria** contains the following fields:

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
Job Name	Used to search for scheduler job having a particular name.	No
Detailed Description	Used to search for scheduler job having the particular description.	No

- **Search Results** - On clicking the **Search** button, the search results appear based on the specified search criteria. The **Search Results** section contains the following columns:

Column Name	Column Description
Job Name	Displays the name of the scheduler job.
	Note: It has link. On clicking the link, the **Scheduler Job** screen appears where you can view the details of the respective scheduler job.
Detailed Description	Displays the detail description of the scheduler job.
Edit	On clicking the **Edit** ( ) icon, the **Scheduler Job** screen appears where you can edit the details of the existing scheduler job.
Delete	On clicking the **Delete** ( ) icon, you can delete the existing scheduler job.
Submit	On clicking the **Submit** icon, you can submit the existing scheduler job for execution.

Related Topics

For more information on...	See...
How to search for a scheduler job	Searching for a Scheduler Job on page 1052
How to define a scheduler job	Defining a Scheduler Job on page 1053
How to view the scheduler job details	Viewing a Scheduler Job Details on page 1055
How to edit a scheduler job	Editing a Scheduler Job on page 1055
How to delete a scheduler job	Deleting a Scheduler Job on page 1056
How to submit a scheduler job	Submitting a Scheduler Job on page 1057

Searching for a Scheduler Job

Prerequisites

To search a scheduler job, you should have:

- Scheduler job defined in the application.

Procedure

To search a scheduler job:

1. Click the **Menu** link in the **Application** toolbar.
   A list appears.
2. From the **Main** menu, select**Batch Scheduler** and then click the **Scheduler Job** sub-menu.
   The **Scheduler Job** screen appears.
3. Enter the search criteria in the **Search Scheduler Job** zone.

Copyright © 2024, Oracle and/or its affiliates.

> **Note:** ORMB search engine supports wildcard search, where you can substitute the percentage (%) symbol as a stand in for any word or letter in a search criteria. You can use the '%' wildcard character in all input fields except the date and ID fields. The '%' wildcard character is suffixed automatically at the end of the partial search criteria. Therefore, you may or may not specify the wildcard character at the end of the partial search criteria. However, you have to prefix the wildcard character manually wherever required.

4. Click **Search** button.
   A list of scheduler job that meet the search criteria appears in the **Search Results** section.

Related Topics

For more information on...	See...
Search Scheduler Job zone	Search Scheduler Job on page 1051
How to define a scheduler job	Defining a Scheduler Job on page 1053
How to view the scheduler job	Viewing a Scheduler Job Details on page 1055
How to edit a scheduler job	Editing a Scheduler Job on page 1055
How to delete a scheduler job	Deleting a Scheduler Job on page 1056
How to submit a scheduler job	Submitting a Scheduler Job on page 1057

Defining a Scheduler Job

Prerequisites

To add a scheduler job, you should have:

- Program name or chain name defined in the system.
- Schedule name defined in the system.

Procedure

To add a scheduler job:

1. Click the **Menu** link in the **Application** toolbar.
   A list appears.
2. From the **Main** menu, select**Batch Scheduler** and then click the **Scheduler Job** sub-menu.
   The **Scheduler Job** screen appears.
3. Click **Add** link in the page title area of the **Search Scheduler Job** screen.
   The **Scheduler Job** screen appears. It contains the following fields:

   j

Field Name	Field Description	Mandatory (Yes or No)
Job Name	Used to specify the name of scheduler job.	Yes

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: You should prefix the job name with **CM** (customer modification) while defining the job on the deployed environment. For example: The changes performed during and after the installation of the product. You should prefix the job name with **C1** while defining the job on the development environment. For example: The changes performed on customer modification data.	
	The valid formats are: • C1_BILL • C1BILL • CM_BILL • CMBILL The invalid formats are: • C1-BILL • C1 BILL • CM-BILL • CM BILL	
Detailed Description	Used to specify the description of the scheduler job.	No
Program Name	Used to indicate the name of scheduler program.	Yes (Conditional)
	Note: The **Search** (🔍) icon appears corresponding to this field. On clicking the **Search** icon, the **Scheduler Program Search** window appears.	**Note:** This data is required if you do not specify data in the **Chain Name** field.
Chain Name	Used to indicate the name of scheduler chain.	Yes (Conditional)
	Note: The **Search** (🔍) icon appears corresponding to this field. On clicking the **Search** icon, the **Scheduler Chain Search** window appears.	**Note:** This data is required if you do not specify data in the **Program Name** field.
Schedule Name	Used to indicate the name of scheduler name.	No
	Note: The **Search** (🔍) icon appears corresponding to this field. On clicking the **Search** icon, the **Scheduler Schedule Search** window appears.	

4. Click **Save**.
   The scheduler job is added.

Copyright © 2024, Oracle and/or its affiliates.

Related Topics

For more information on...	See...
Search Scheduler Job zone	Search Scheduler Job on page 1051
How to search for a scheduler job	Searching for a Scheduler Job on page 1052
How to view the scheduler job details	Viewing a Scheduler Job Details on page 1055
How to edit a scheduler job	Editing a Scheduler Job on page 1055
How to delete a scheduler job	Deleting a Scheduler Job on page 1056
How to submit a scheduler job	Submitting a Scheduler Job on page 1057

Viewing a Scheduler Job Details

Procedure

To view a scheduler job:

1. Search for the scheduler job in the **Search Scheduler Job** zone of the **Scheduler Job** screen.

   Note: ORMB search engine supports wildcard search, where you can substitute the percentage (%) symbol as a stand in for any word or letter in a search criteria. You can use the '%' wildcard character in all input fields except the date and ID fields. The '%' wildcard character is suffixed automatically at the end of the partial search criteria. Therefore, you may or may not specify the wildcard character at the end of the partial search criteria. However, you have to prefix the wildcard character manually wherever required.

2. In the **Search Result** section, click the link in the **Job Name** column whose details you want to view.
   The **Scheduler Job** screen appears.
3. View the details of scheduler job in the **Scheduler Job** screen.

Related Topics

For more information on...	See...
Search Scheduler Job zone	Search Scheduler Job on page 1051
How to search for a scheduler job	Searching for a Scheduler Job on page 1052
How to define a scheduler job	Defining a Scheduler Job on page 1053
How to edit a scheduler job	Editing a Scheduler Job on page 1055
How to delete a scheduler job	Deleting a Scheduler Job on page 1056
How to submit a scheduler job	Submitting a Scheduler Job on page 1057

Editing a Scheduler Job

Procedure

To edit a scheduler job:

1. Search for the scheduler job in the **Scheduler Job** screen whose details you want to edit.
2. In the **Search Results** section, click the **Edit** (✎) icon in the **Edit** column corresponding to the scheduler job whose details you want to edit.
   The **Scheduler Job** screen appears. It contains the following fields:

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
Job Name	Indicates the name of scheduler job.	No
Detailed Description	Used to specify the description of the scheduler program.	No
Program Name	Used to indicate the name of scheduler program.	Yes (Conditional)
	Note: The **Search** (🔍) icon appears corresponding to this field. On clicking the **Search** icon, the **Scheduler Program Search** window appears.	**Note:** This data is required if you do not specify data in the **Chain Name** field.
Chain Name	Used to indicate the name of scheduler chain.	Yes (Conditional)
	Note: The **Search** (🔍) icon appears corresponding to this field. On clicking the **Search** icon, the **Scheduler Chain Search** window appears.	**Note:** This data is required if you do not specify data in the **Program Name** field.
Schedule Name	Used to indicate the name of scheduler name.	No
	Note: The **Search** (🔍) icon appears corresponding to this field. On clicking the **Search** icon, the **Scheduler Schedule Search** window appears.	

3. Click **Save**.
   The scheduler job is modified.

Related Topics

For more information on...	See...
Search Scheduler Job zone	Search Scheduler Job on page 1051
How to search for a scheduler job	Searching for a Scheduler Job on page 1052
How to define a scheduler job	Defining a Scheduler Job on page 1053
How to view the scheduler job details	Viewing a Scheduler Job Details on page 1055
How to delete a scheduler job	Deleting a Scheduler Job on page 1056
How to submit a scheduler job	Submitting a Scheduler Job on page 1057

Deleting a Scheduler Job

Procedure

To delete a scheduler job:

1. Search for the scheduler job in the **Search Scheduler Job** zone of the **Scheduler Job** screen.

Copyright © 2024, Oracle and/or its affiliates.

> **Note:** ORMB search engine supports wildcard search, where you can substitute the percentage (%) symbol as a stand in for any word or letter in a search criteria. You can use the '%' wildcard character in all input fields except the date and ID fields. The '%' wildcard character is suffixed automatically at the end of the partial search criteria. Therefore, you may or may not specify the wildcard character at the end of the partial search criteria. However, you have to prefix the wildcard character manually wherever required.

2. In the **Search Result** section, click the **Delete** (🗑) icon in the **Delete** column corresponding to the scheduler job which you want to delete.
   A message appears confirming whether you want to delete the scheduler job.
3. Click **Ok**.
   The scheduler job is deleted.

Related Topics

For more information on...	See...
Search Scheduler Job zone	Search Scheduler Job on page 1051
How to search for a scheduler job	Searching for a Scheduler Job on page 1052
How to define a scheduler job	Defining a Scheduler Job on page 1053
How to view the scheduler job details	Viewing a Scheduler Job Details on page 1055
How to edit a scheduler job	Editing a Scheduler Job on page 1055
How to submit a scheduler job	Submitting a Scheduler Job on page 1057

Submitting a Scheduler Job

Prerequisites

To submit a scheduler job, you should have:

- Scheduler job defined in the application.

Procedure

To submit a scheduler job:

1. Search for the scheduler job in the **Scheduler Job** screen whose details you want to submit.

> **Note:** ORMB search engine supports wildcard search, where you can substitute the percentage (%) symbol as a stand in for any word or letter in a search criteria. You can use the '%' wildcard character in all input fields except the date and ID fields. The '%' wildcard character is suffixed automatically at the end of the partial search criteria. Therefore, you may or may not specify the wildcard character at the end of the partial search criteria. However, you have to prefix the wildcard character manually wherever required.

2. In the **Search Results** section, click the **Submit** icon in the **Submit Job** column corresponding to the scheduler job which you want to submit.
   A message appears confirming whether you want to submit the scheduler job.
3. Click **Ok**.
   The scheduler job is submitted.

Related Topics

For more information on...	See...
Search Scheduler Job zone	Search Scheduler Job on page 1051
How to search for a scheduler job	Searching for a Scheduler Job on page 1052

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
How to define a scheduler job	Defining a Scheduler Job on page 1053
How to view the scheduler job details	Viewing a Scheduler Job Details on page 1055
How to edit a scheduler job	Editing a Scheduler Job on page 1055
How to delete a scheduler job	Deleting a Scheduler Job on page 1056

Scheduler Job Monitor

The **Scheduler Job Monitor** screen allows you to search the submitted job using various search criteria. It contains the following zone:

- Search Scheduler Job on page 1058

Related Topics

For more information on...	See...
How to search for a scheduler job monitor	Searching for a Submitted Scheduler Job on page 1059
How to view the scheduler job	Viewing a Scheduler Job Details on page 1060
How to cleanup a scheduler job	
How to cancel a scheduler job	Cancelling a Scheduler Job on page 1060
How to restart a scheduler job	Restarting a Scheduler Job on page 1061

Search Scheduler Job

The **Search Scheduler Job** zone allows you to search for the submitted job using various search criteria. It contains the following sections:

- **Search Criteria** - The **Search Criteria** contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Scheduler Job Status	Used to search for scheduler job with a particular status. The valid values are: • Completed • In Progress	Yes
Run Start Date	Used to search scheduler job which are running from a particular date onwards.	No
Run End Date	Used to search scheduler job which are running till a particular date.	No

- **Search Results** - On clicking the **Search** button, the search results appear based on the specified search criteria. The **Search Results** section contains the following columns:

Column Name	Column Description
Job Name	Displays the name of the job.

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
Job Status	Displays the status of the job.
Run Start Date	Displays the date from when the scheduler job is executed.
Run End Date	Displays the date till when the scheduler job is executed.
Chain Name	Displays the chain name of the respective job.
Current Running Step	Displays the current running steps of the job.

The **Search Results** section also contains the following buttons:

- **Details** -

  On clicking **Details** button, the **Scheduler Job Details** screen appears which displays the detailed information of the respective job.
- **Cancel**

  - On clicking **Cancel** button, the respective job gets cancel.

  Note: The **Cancel** button is disable when the scheduler job status is completed.
- **Restart**

  - On clicking **Restart** button, the respective failed job get restarted.

Related Topics

For more information on...	See...
How to search for a scheduler job	Searching for a Submitted Scheduler Job on page 1059
How to view the scheduler job	Viewing a Scheduler Job Details on page 1060
How to cleanup a scheduler job	
How to cancel a scheduler job	Cancelling a Scheduler Job on page 1060
How to restart a scheduler job	Restarting a Scheduler Job on page 1061

Searching for a Submitted Scheduler Job

Prerequisites

To search a submitted scheduler job, you should have:

- Scheduler job defined in the application.

Procedure

To search a submitted scheduler job:

1. Click the **Menu** link in the **Application** toolbar.
   A list appears.
2. From the **Main** menu, select**Batch Scheduler** and then click the **Scheduler Job Monitor** sub-menu.
   The **Scheduler Job Monitor** screen appears.
3. Enter the search criteria in the **Search Scheduler Job** zone.
4. Click **Search** button.
   A list of scheduler job that meet the search criteria appears in the **Search Results** section.

Related Topics

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
Search Scheduler Job Zone	Search Scheduler Job on page 1058
How to view the scheduler job	Viewing a Scheduler Job Details on page 1060
How to cleanup a scheduler job	
How to cancel a scheduler job	Cancelling a Scheduler Job on page 1060
How to restart a scheduler job	Restarting a Scheduler Job on page 1061

Viewing a Scheduler Job Details

Procedure

To view a scheduler job details:

1. Search for the scheduler job in the **Search Scheduler Job** zone of the **Scheduler Job Monitor** screen.
2. In the **Search Result** section, click the **Details** button corresponding to the job whose details you want to view.
   The **Scheduler Job Details** screen appears.
3. View the details of scheduler job in the **Scheduler Job Details** screen.

Related Topics

For more information on...	See...
Search Scheduler Job Zone	Search Scheduler Job on page 1058
How to search for a scheduler job	Searching for a Submitted Scheduler Job on page 1059
How to cleanup a scheduler job	
How to cancel a scheduler job	Cancelling a Scheduler Job on page 1060
How to restart a scheduler job	Restarting a Scheduler Job on page 1061

Cancelling a Scheduler Job

Procedure

To cancel a scheduler job:

1. Search for the scheduler job in the **Search Scheduler Job** zone of the **Scheduler Job Monitor** screen.
2. In the **Search Result** section, click the **Cancel** button corresponding to the job which you want to cancel.

   Note: The **Cancel** button is enable only when the status of scheduler job is **In Progress** status.

   A message appears confirming whether you want to cancel the job.
3. Click **Ok**.
   The scheduler job is cancelled.

Related Topics

For more information on...	See...
Search Scheduler Job Zone	Search Scheduler Job on page 1058
How to search for a scheduler job	Searching for a Submitted Scheduler Job on page 1059
How to cleanup a scheduler job	

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
How to view the scheduler job	Viewing a Scheduler Job Details on page 1060
How to restart a scheduler job	Restarting a Scheduler Job on page 1061

Restarting a Scheduler Job

Procedure

To restart a scheduler job:

1. Search for the scheduler job in the **Search Scheduler Job** zone of the **Scheduler Job Monitor** screen.
2. In the **Search Result** section, click the **Restart** button corresponding to the job which you want to restart.
   A message appears confirming whether you want to restart the job.
3. Click **Ok**.
   The scheduler job is restarted.

Related Topics

For more information on...	See...
Search Scheduler Job Zone	Search Scheduler Job on page 1058
How to search for a scheduler job	Searching for a Submitted Scheduler Job on page 1059
How to cleanup a scheduler job	
How to view the scheduler job	Viewing a Scheduler Job Details on page 1060
How to cancel a scheduler job	Cancelling a Scheduler Job on page 1060

Scheduler Job Monitor Dashboard

The **Scheduler Job Monitor Dashboard** screen allows you to view in progress and failed scheduler jobs. It also allows you to search executed and upcoming scheduler jobs using various search criteria. It contains the following zones:

- In Progress Scheduler Jobs on page 1061
- Failed Scheduler Jobs on page 1062
- Completed Scheduler Jobs on page 1062
- Upcoming Scheduler Jobs on page 1063
-

In Progress Scheduler Jobs

The **In Progress Scheduler Jobs** zone allows you to view currently executed scheduler jobs. It contains the following columns:

Column Name	Column Description
Job Name	Displays the name of the currently executed scheduler job.
Run Duration	Displays the duration required during the execution of respective scheduler job.
Current Running Step	Displays the currently running step of the executed scheduler job.

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
View Details	On clicking **View** button, the **Batch Job Detailed Description** screen appears where you can see detail information of the respective scheduler job.

Related Topics

For more information on...	See...
Failed Scheduler Jobs Zone	Failed Scheduler Jobs on page 1062
Completed Scheduler Jobs Zone	Completed Scheduler Jobs on page 1062
Upcoming Scheduler Jobs Zone	Upcoming Scheduler Jobs on page 1063

Failed Scheduler Jobs

The **Failed scheduler Jobs** zone allows you to view scheduler jobs which are partially executed and the failed jobs at particular step. It contains the following columns:

Column Name	Column Description
Job Name	Displays the name of the failed scheduler job.
Run Start Date	Displays the date of executed scheduler job which is running from a particular date onwards.
Run Duration	Displays the duration required during the execution of respective scheduler job.
View Details	On clicking **View** button, the **Batch Job Detailed Description** screen appears where you can see detail information of the respective scheduler job.

Related Topics

For more information on...	See...
In Progress Scheduler Jobs Zone	In Progress Scheduler Jobs on page 1061
Completed Scheduler Jobs Zone	Completed Scheduler Jobs on page 1062
Upcoming Scheduler Jobs Zone	Upcoming Scheduler Jobs on page 1063

Completed Scheduler Jobs

The **Completed Scheduler Jobs** zone allows you to search for the executed scheduler jobs using various search criteria. It contains the following sections:

- **Search Criteria** - The **Search Criteria** contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Job Status	Used to search for completed scheduler job with a particular status. The valid values are: • Succeeded	No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	• Stopped • Failed	
Job Name	Used to search for completed scheduler job having a particular name.	No
Start Date	Used to search completed scheduler job which are running from a particular date onwards.	No
End Date	Used to search completed scheduler job which are running till a particular date.	No

- **Search Results** - On clicking the **Search** button, the search results appear based on the specified search criteria. The **Search Results** section contains the following columns:

Column Name	Column Description
Job Name	Displays the name of the completed scheduler job.
Job Status	Displays the status the completed scheduler job. The valid values are: • Succeeded • Stopped • Failed
Total Steps	Displays total number of steps included in the scheduler job. **Note:** By default, the total step for programs is 1.
Failed Steps	Displays the number of failed steps out of total number of steps in a scheduler job.
Run Start Date	Displays the date of completed scheduler job which is running from a particular date onwards.
Submission User	
View Details	On clicking **View** button, the **Batch Job Detailed Description** screen appears where you can see detail information of the respective completed scheduler job.

Related Topics

For more information on...	See...
In Progress Scheduler Jobs Zone	In Progress Scheduler Jobs on page 1061
Failed Scheduler Jobs Zone	Failed Scheduler Jobs on page 1062
Upcoming Scheduler Jobs Zone	Upcoming Scheduler Jobs on page 1063

Upcoming Scheduler Jobs

The **Upcoming Scheduler Jobs** zone allows you to search for the upcoming scheduler jobs using search criteria. It contains the following sections:

- **Search Criteria** - The **Search Criteria** contains the following field:

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
Schedule Type	Used to search for upcoming scheduled job with a particular status. The valid values are: • Next 5 Jobs Schedule - Used to view the list of 5 upcoming scheduled jobs. • Next Day Schedule - Used to view the list of scheduled jobs which are executing in next 24 hours.	Yes

- **Search Results** - On clicking the **Search** button, the search results appear based on the specified search criteria. The **Search Results** section contains the following columns:

Column Name	Column Description
Job Name	Displays the name of the upcoming scheduled jobs.
Program Name	Displays the name of the program which is in the scheduled job. In case of chain jobs, column value will be null.
Next Action Date	Displays the scheduled date on which the upcoming scheduled job is automatically triggered.
Schedule Name	Displays the name of the schedule which is in the scheduled job. It has link. On clicking the link, the **Scheduler Schedule** screen appears.

Related Topics

For more information on...	See...
In Progress Scheduler Jobs Zone	In Progress Scheduler Jobs on page 1061
Failed Scheduler Jobs Zone	Failed Scheduler Jobs on page 1062
Completed Scheduler Jobs Zone	Completed Scheduler Jobs on page 1062

Scheduler Job Details

The **Scheduler Job Details** screen allows you to view in progress, failed, completed, and upcoming scheduler job details. It contains the following zones:

- In Progress Scheduler Jobs Details on page 1065
- Failed Scheduler Jobs Details on page 1065
- Completed Scheduler Jobs Details on page 1066
- Scheduler Job Details Zone on page 1066

Related Topics

For more information on...	See...
How to view the scheduler job details	Viewing a Scheduler Job Details on page 1067

Copyright © 2024, Oracle and/or its affiliates.

In Progress Scheduler Jobs Details

The **In Progress Scheduler Jobs** zone allows you to view currently executed scheduler jobs using various search criteria. It contains the following columns:

Column Name	Column Description
Job Name	Displays the name of the currently executed batch job.
Job Status	Displays the current status of the job.
Run Start Date	Displays the date of executed job which is running from a particular date onwards.
Run Duration	Displays the duration required during the execution of respective job.
Program Name	Displays the name of the program which is in the current scheduled job.
Chain Name	Displays the name of the chain.
Current Running Step	Displays the current running step of the executed job.
Restart Job	This button is disabled when scheduler job is in progress.
Cancel	On clicking **Cancel** button, the current job get cancelled.

Related Topics

For more information on...	See...
Failed Scheduler Jobs details zone	Failed Scheduler Jobs Details on page 1065
Completed Scheduler Jobs details zone	Completed Scheduler Jobs Details on page 1066
Scheduler Job Details zone	Scheduler Job Details Zone on page 1066
How to view the scheduler job details	Viewing a Scheduler Job Details on page 1067

Failed Scheduler Jobs Details

The **Failed Scheduler Jobs Details** zone allows you to view failed jobs. It contains the following columns:

Column Name	Column Description
Job Name	Displays the name of the failed job.
Job Status	Displays the status of the job.
Run Start Date	Displays the date when the failed job start to execute.
Run Duration	Displays the duration required during the execution of respective job.
Program Name	Displays the name of the program which is in the failed scheduled job.
Chain Name	Displays the name of the chain.
Last Executed Step	Displays the last executed step of the failed job.
Restart Job	On clicking **Restart** button, the failed job gets restarted.
Cancel Job	On clicking **Cancel** button, the failed job get cancelled.

Related Topics

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
In Progress Scheduler Jobs details zone	In Progress Scheduler Jobs Details on page 1065
Completed Scheduler Jobs details zone	Completed Scheduler Jobs Details on page 1066
Scheduler Job Details zone	Scheduler Job Details Zone on page 1066
How to view the scheduler job details	Viewing a Scheduler Job Details on page 1067

Completed Scheduler Jobs Details

The **Completed Scheduler Jobs Details** zone allows you to view completed jobs. It contains the following columns:

Column Name	Column Description
Job Name	Displays the name of the completed job.
Job Status	Displays the status of the job.
Total Steps	Displays the total number of step of the completed job.
Failed Steps	Displays the count of failed step of the completed job.
Run Start Date	Displays the date when the completed job start to execute.
Run Duration	Displays the duration required during the execution of respective job.
Program Name	Displays the name of the program which is in the completed scheduled job.
Chain Name	Displays the name of the chain.
Restart Job	On clicking **Restart** button, the job gets restarted.
	Note: The **Restart** button will be enable only if the job is in **Failed** status.
Cancel Job	On clicking **Cancel** button, the job get cancelled.
	Note: The **Cancel** button will be enable only if the job is in **Failed** status.

Related Topics

For more information on...	See...
In Progress Scheduler Jobs details zone	In Progress Scheduler Jobs Details on page 1065
Failed Scheduler Jobs details zone	Failed Scheduler Jobs Details on page 1065
Scheduler Job Details zone	Scheduler Job Details Zone on page 1066
How to view the scheduler job details	Viewing a Scheduler Job Details on page 1067

Scheduler Job Details Zone

The **Scheduler Jobs Details** zone allows you to view scheduler jobs details. It contains the following columns:

Column Name	Column Description
Step Name	Displays the name of the step in the scheduler job.

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description
Program Name	Displays the name of the program which is in the current scheduler job.
Job Status	Displays the current status of the scheduler job.
Run Duration	Displays the duration required during the execution of respective job.
Error Details	Displays the detail of error occur during execution of the scheduler job.
Batch Control	Displays the name of the batch control.
Processed Records	Displays the number of processed records.
Failed Records	Displays the number of failed records.

Related Topics

For more information on...	See...
In Progress Scheduler Jobs details zone	In Progress Scheduler Jobs Details on page 1065
Failed Scheduler Jobs details zone	Failed Scheduler Jobs Details on page 1065
Completed Scheduler Jobs details zone	Completed Scheduler Jobs Details on page 1066
How to view the scheduler job details	Viewing a Scheduler Job Details on page 1067

Viewing a Scheduler Job Details

Procedure

To view a scheduler job:

1. In the **In Progress Scheduler Job**, **Failed Scheduler Job**, and **Completed Scheduler Job** zone, click the **Broadcast** icon corresponding to the job in the whose details you want to view.
   The **Batch Job Details** zone appears.
2. View the details of scheduler job in the **Batch Job Details** zone.

Related Topics

For more information on...	See...
In Progress Scheduler Jobs Details zone	In Progress Scheduler Jobs Details on page 1065
Failed Scheduler Jobs Details zone	Failed Scheduler Jobs Details on page 1065
Completed Scheduler Jobs Details zone	Completed Scheduler Jobs Details on page 1066
Scheduler Job Details zone	Scheduler Job Details Zone on page 1066

Scheduler Program

The **Scheduler Program** screen allows you to search a program using various search criteria. It also allows you to define, view, edit, and delete a scheduler program. It contains the following zone:

- Search Scheduler Program on page 1068

Related Topics

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
How to search for a scheduler program	Searching for a Scheduler Program on page 1069
How to define a scheduler program	Defining a Scheduler Program on page 1069
How to view the scheduler program details	Viewing a Scheduler Program Details on page 1072
How to edit a scheduler program	Editing a Scheduler Program on page 1073
How to delete a scheduler program	Deleting a Scheduler Program on page 1075

Search Scheduler Program

The **Search Scheduler Program** zone allows you to search for a scheduler program using various search criteria. It contains the following sections:

- **Search Criteria** - The **Search Criteria** contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Program Name	Used to search for scheduler program having a particular name.	No
Detailed Description	Used to search for scheduler program having the particular description.	No

- **Search Results** - On clicking the **Search** button, the search results appear based on the specified search criteria. The **Search Results** section contains the following columns:

Column Name	Column Description
Program Name	Displays the name of the scheduler program. **Note:** It has link. On clicking the link, the **Scheduler Program** screen appears where you can view the details of the respective scheduler program.
Detailed Description	Displays the detail description of the scheduler program.
Edit	On clicking the **Edit** ( ) icon, the **Scheduler Program** screen appears where you can edit the details of the existing scheduler program.
Delete	On clicking the **Delete** ( ) icon, you can delete the existing scheduler program.

Related Topics

For more information on...	See...
How to search for a scheduler program	Searching for a Scheduler Program on page 1069
How to define a scheduler program	Defining a Scheduler Program on page 1069
How to view the scheduler program details	Viewing a Scheduler Program Details on page 1072
How to edit a scheduler program	Editing a Scheduler Program on page 1073
How to delete a scheduler program	Deleting a Scheduler Program on page 1075

Copyright © 2024, Oracle and/or its affiliates.

Searching for a Scheduler Program

Prerequisites

To search a scheduler program, you should have:

- Scheduler program defined in the application.

Procedure

To search a scheduler program:

1. Click the **Menu** link in the **Application** toolbar.
   A list appears.
2. From the **Main** menu, select**Batch Scheduler** and then click the **Scheduler Program** sub-menu.
   The **Scheduler Program** screen appears.
3. Enter the search criteria in the **Search Scheduler Program** zone.

   Note: ORMB search engine supports wildcard search, where you can substitute the percentage (%) symbol as a stand in for any word or letter in a search criteria. You can use the '%' wildcard character in all input fields except the date and ID fields. The '%' wildcard character is suffixed automatically at the end of the partial search criteria. Therefore, you may or may not specify the wildcard character at the end of the partial search criteria. However, you have to prefix the wildcard character manually wherever required.
4. Click **Search** button.
   A list of scheduler program that meet the search criteria appears in the **Search Results** section.

Related Topics

For more information on...	See...
Search Scheduler Program zone	Search Scheduler Program on page 1068
How to define a scheduler program	Defining a Scheduler Program on page 1069
How to view the scheduler program details	Viewing a Scheduler Program Details on page 1072
How to edit a scheduler program	Editing a Scheduler Program on page 1073
How to delete a scheduler program	Deleting a Scheduler Program on page 1075

Defining a Scheduler Program

Prerequisites

To add a scheduler program, you should have:

- User Id defined in the system.
- Batch control defined in the system.

Procedure

To add a scheduler program:

1. Click the **Menu** link in the **Application** toolbar.
   A list appears.
2. From the **Main** menu, select**Batch Scheduler** and then click the **Scheduler Program** sub-menu.
   The **Scheduler Program** screen appears.
3. Click **Add** link in the page title area of the **Search Scheduler Program** screen.
   The **Scheduler Program** screen appears. It contains the following fields:

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
Program Name	Used to specify the name of scheduler program. **Note:** You should prefix the program name with **CM** (customer modification) while defining the program on the deployed environment. For example: The changes performed during and after the installation of the product. You should prefix the program name with **C1** while defining the program on the development environment. For example: The changes performed on customer modification data. The valid formats are: • C1_BILL • C1BILL • CM_BILL • CMBILL The invalid formats are: • C1-BILL • C1 BILL • CM-BILL • CM BILL	Yes
Detailed Description	Used to specify the description of the scheduler program.	No
Batch Control	Used to indicate the code which executes the logic and associated with the background process. **Note:** The **Search** (🔍) icon appears corresponding to this field. On clicking the **Search** icon, the **Batch Control Search** window appears.	Yes
User ID	Used to indicate the user id of the user. **Note:** The **Search** (🔍) icon appears corresponding to this field. On clicking the **Search** icon, the **User Search** window appears.	Yes
Thread Count	Used to specify the total number of threads that you want to schedule in parallel.	No
Invoke Function	Used to indicate whether you want to invoke function for batch business date or not.	No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: If the **Invoke Function** checkbox is selected, then the batch business date accepts the string value instead of date value.	
Batch Business Date	Used to specify the date for background process which is used in the processing of the process. For example, if the job is scheduled using business date then it determine which program or chain should be scheduled. **Note:** If you not specify, then it is set to system date.	No
Maximum Timeout Minutes	Used to specify the time duration for overriding each background process.	No
Batch Rerun Number	Used to specify run number of the background processes. **Note:** You should specify this field only if you want to download the information of run history of the background process.	No
Thread Pool Name	Used to specify the thread pool on which you want to execute the batch.	No
Run Status	Indicates the status of the batch run. The valid values are: • Complete • Error • In Progress • Thread Ready	No
Raise Error	Used to specify whether you want to raise the error. The valid values are: • True • False **Note:** If you specify the**Run Status** as **Error** and the **Raise Error** as **True**, then the system will throw an error and end the job.	No
Thread Notification	Used to specify whether you want send the notification through email if the threads are failed during execution or not. The valid values are: • True • False	No

The **Scheduler Program** screen also contains the following columns:

Copyright © 2024, Oracle and/or its affiliates.

Column Name	Column Description	Mandatory (Yes or No)
Parameter Name	Displays the name of the parameter which is associated with the respective batch.	Not applicable
Description	Displays the description of the parameter.	Not applicable
Parameter Value	Used to specify the value of the parameter.	No
Required	Indicated that whether the parameter is required for batch or not.	Not applicable

4. Click **Save**.
   The scheduler job is added.

Related Topics

For more information on...	See...
Search Scheduler Program zone	Search Scheduler Program on page 1068
How to search for a scheduler program	Searching for a Scheduler Program on page 1069
How to view the scheduler program details	Viewing a Scheduler Program Details on page 1072
How to edit a scheduler program	Editing a Scheduler Program on page 1073
How to delete a scheduler program	Deleting a Scheduler Program on page 1075

Viewing a Scheduler Program Details

Procedure

To view a scheduler program:

1. Search for the scheduler program in the **Search Scheduler Program** zone of the **Scheduler Program** screen.

   Note: ORMB search engine supports wildcard search, where you can substitute the percentage (%) symbol as a stand in for any word or letter in a search criteria. You can use the '%' wildcard character in all input fields except the date and ID fields. The '%' wildcard character is suffixed automatically at the end of the partial search criteria. Therefore, you may or may not specify the wildcard character at the end of the partial search criteria. However, you have to prefix the wildcard character manually wherever required.

2. In the **Search Result** section, click the link in the **Program Name** column whose details you want to view.
   The **Scheduler Program** screen appears.
3. View the details of scheduler program in the **Scheduler Program** screen.

Related Topics

For more information on...	See...
Search Scheduler Program zone	Search Scheduler Program on page 1068
How to search for a scheduler program	Searching for a Scheduler Program on page 1069
How to define a scheduler program	Defining a Scheduler Program on page 1069
How to edit a scheduler program	Editing a Scheduler Program on page 1073
How to delete a scheduler program	Deleting a Scheduler Program on page 1075

Copyright © 2024, Oracle and/or its affiliates.

Editing a Scheduler Program

Procedure

To edit a scheduler program:

1. Search for the scheduler program in the **Scheduler Program** screen whose details you want to edit.
2. In the **Search Results** section, click the **Edit** (✎) icon in the **Edit** column corresponding to the scheduler program whose details you want to edit.
   The **Scheduler Program** screen appears. It contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Program Name	Displays the name of scheduler program.	Not applicable
Detailed Description	Used to specify the description of the scheduler program.	No
Batch Control	Used to indicate the code which executes the logic and associated with the background process. **Note:** The **Search** (🔍) icon appears corresponding to this field. On clicking the **Search** icon, the **Batch Control Search** window appears.	No
User ID	Used to indicate the user id of the user. **Note:** The **Search** (🔍) icon appears corresponding to this field. On clicking the **Search** icon, the **User Search** window appears.	No
Thread Count	Used to specify the total number of threads that you want to schedule in parallel.	No
Invoke Function	Used to indicate whether you want to invoke function for batch business date or not. **Note:** If the **Invoke Function** checkbox is selected, then the batch business date accepts the string value instead of date value.	No
Batch Business Date	Used to specify the date for background process which is used in the processing of the process. For example, if the job is scheduled using business date then it determine which program or chain should be scheduled. **Note:** If you not specify, then it is set to system date.	No
Maximum Timeout Minutes	Used to specify the time duration for overriding each background process.	No
Batch Rerun Number	Used to specify run number of the background processes.	No

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
	Note: You should specify this field only if you want to download the information of run history of the background process.	
Thread Pool Name	Used to specify the thread pool on which you want to execute the batch.	No
Run Status	Indicates the status of the batch run. The valid values are: • Complete • Error • In Progress • Thread Ready	Not applicable
Raise Error	Used to specify whether you want to raise the error. The valid values are: • True • False **Note:** If you specify the**Run Status** as **Error** and the **Raise Error** as **True**, then the system will throw an error and end the job.	No
Thread Notification	Used to specify whether you want send the notification through email if the threads are failed during execution or not. The valid values are: • True • False	No

The **Scheduler Program** screen also contains the following columns:

Column Name	Column Description	Mandatory (Yes or No)
Parameter Name	Displays the name of the parameter which is associated with the respective batch.	Not applicable
Description	Displays the description of the parameter.	Not applicable
Parameter Value	Used to specify the value of the parameter.	No
Required	Indicated that whether the parameter is required for batch or not.	Not applicable

3. Click **Save**.
   The scheduler program is modified.

Related Topics

For more information on...	See...
Search Scheduler Program zone	Search Scheduler Program on page 1068
How to search for a scheduler program	Searching for a Scheduler Program on page 1069

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
How to define a scheduler program	Defining a Scheduler Program on page 1069
How to view the scheduler program details	Viewing a Scheduler Program Details on page 1072
How to delete a scheduler program	Deleting a Scheduler Program on page 1075

Deleting a Scheduler Program

Procedure

To delete a scheduler program:

1. Search for the scheduler program in the **Search Scheduler Program** zone of the **Scheduler Program** screen.

   Note: ORMB search engine supports wildcard search, where you can substitute the percentage (%) symbol as a stand in for any word or letter in a search criteria. You can use the '%' wildcard character in all input fields except the date and ID fields. The '%' wildcard character is suffixed automatically at the end of the partial search criteria. Therefore, you may or may not specify the wildcard character at the end of the partial search criteria. However, you have to prefix the wildcard character manually wherever required.

2. In the **Search Result** section, click the **Delete** (🗑) icon in the **Delete** column corresponding to the scheduler program which you want to delete.
   A message appears confirming whether you want to delete the scheduler program.
3. Click **OK**.
   The scheduler program is deleted.

   Note: You can delete a scheduler program only if the scheduler program is not used in the any other jobs.

Related Topics

For more information on...	See...
Search Scheduler Program zone	Search Scheduler Program on page 1068
How to search for a scheduler program	Searching for a Scheduler Program on page 1069
How to define a scheduler program	Defining a Scheduler Program on page 1069
How to view the scheduler program details	Viewing a Scheduler Program Details on page 1072
How to edit a scheduler program	Editing a Scheduler Program on page 1073

Scheduler Schedule

The **Scheduler Schedule** screen allows you to search a schedule using various search criteria. It also allows you to define, view, edit, and delete a scheduler schedule. It contains the following zone:

- Search Scheduler Schedule on page 1076

Related Topics

For more information on...	See...
How to search for a scheduler schedule	Searching for a Scheduler Schedule on page 1077
How to define a scheduler schedule	Defining a Scheduler Schedule on page 1077

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
How to view the scheduler schedule details	Viewing a Scheduler Schedule Details on page 1081
How to edit a scheduler schedule	Editing a Scheduler Schedule on page 1081
How to delete a scheduler schedule	Deleting a Scheduler Schedule on page 1084

Search Scheduler Schedule

The **Search Scheduler Schedule** zone allows you to search for a scheduler schedule using various search criteria. It contains the following sections:

- **Search Criteria** - The **Search Criteria** contains the following fields:

Field Name	Field Description	Mandatory (Yes or No)
Schedule Name	Used to search for scheduler schedule having a particular name.	No
Detailed Description	Used to search for scheduler schedule having the particular description.	No

- **Search Results** - On clicking the **Search** button, the search results appear based on the specified search criteria. The **Search Results** section contains the following columns:

Column Name	Column Description
Schedule Name	Displays the name of the scheduler schedule.
	Note: It has link. On clicking the link, the **Scheduler Schedule** screen appears where you can view the details of the respective scheduler schedule.
Detailed Description	Displays the detail description of the scheduler schedule.
Repeat Interval	Displays the concatenated string of information about the interval of the job scheduling.
Edit	On clicking the **Edit** ( ) icon, the **Scheduler Schedule** screen appears where you can edit the details of the existing scheduler schedule.
Delete	On clicking the **Delete** ( ) icon, you can delete the existing scheduler schedule.

Related Topics

For more information on...	See...
How to search for a scheduler schedule	Searching for a Scheduler Schedule on page 1077
How to define a scheduler schedule	Defining a Scheduler Schedule on page 1077
How to view the scheduler schedule details	Viewing a Scheduler Schedule Details on page 1081
How to edit a scheduler schedule	Editing a Scheduler Schedule on page 1081
How to delete a scheduler schedule	Deleting a Scheduler Schedule on page 1084

Copyright © 2024, Oracle and/or its affiliates.

Searching for a Scheduler Schedule

Prerequisites

To search a scheduler schedule, you should have:

- Scheduler schedule defined in the application.

Procedure

To search a scheduler schedule:

1. Click the **Menu** link in the **Application** toolbar.
   A list appears.
2. From the **Main** menu, select**Batch Scheduler** and then click the **Scheduler Schedule** sub-menu.
   The **Scheduler Schedule** screen appears.
3. Enter the search criteria in the **Search Scheduler Schedule** zone.

   Note: ORMB search engine supports wildcard search, where you can substitute the percentage (%) symbol as a stand in for any word or letter in a search criteria. You can use the '%' wildcard character in all input fields except the date and ID fields. The '%' wildcard character is suffixed automatically at the end of the partial search criteria. Therefore, you may or may not specify the wildcard character at the end of the partial search criteria. However, you have to prefix the wildcard character manually wherever required.
4. Click **Search** button.
   A list of scheduler schedule that meet the search criteria appears in the **Search Results** section.

Related Topics

For more information on...	See...
Search Scheduler Schedule zone	Search Scheduler Schedule on page 1076
How to define a scheduler schedule	Defining a Scheduler Schedule on page 1077
How to view the scheduler schedule details	Viewing a Scheduler Schedule Details on page 1081
How to edit a scheduler schedule	Editing a Scheduler Schedule on page 1081
How to delete a scheduler schedule	Deleting a Scheduler Schedule on page 1084

Defining a Scheduler Schedule

Prerequisites

To add a scheduler schedule, you should have:

- Frequency options defined in the system.

Procedure

To add a scheduler schedule:

1. Click the **Menu** link in the **Application** toolbar.
   A list appears.
2. From the **Main** menu, select**Batch Scheduler** and then click the **Scheduler Schedule** sub-menu.
   The **Scheduler Schedule** screen appears.
3. Click **Add** link in the page title area of the **Search Scheduler Schedule** screen.
   The **Scheduler Schedule** screen appears. It contains the following fields:

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
Schedule Name	Used to specify the name of scheduler schedule. **Note:** You should prefix the schedule name with **CM** (customer modification) while defining the schedule on the deployed environment. For example: The changes performed during and after the installation of the product. You should prefix the schedule name with **C1** while defining the schedule on the development environment. For example: The changes performed on customer modification data. The valid formats are: • C1_BILL • C1BILL • CM_BILL • CMBILL The invalid formats are: • C1-BILL • C1 BILL • CM-BILL • CM BILL	Yes
Detailed Description	Used to specify the description of the scheduler schedule.	No
Repeat Interval	Used to indicate the interval when the system should start the job scheduling.	Yes
Frequency	Used to indicate the frequency of the job recurrence. **Tip:** To select frequency type click the **Generate** button.	Yes

4. To set a time interval, click **Generate** button.
5. Select the frequency option from the drop-down list of **Frequency**.
   It contains the following options:

List Options	Description	Parameter Values
Yearly	Used to set the value of frequency to yearly for the schedule.	Interval - Used to indicate the whether the schedule is scheduled for every year (a value of 1), every other year (a value of 2), or every third year (a value of 3) and so on.

Copyright © 2024, Oracle and/or its affiliates.

List Options	Description	Parameter Values
		Month and Day - Used to indicate the date and month when you want to schedule the schedule. Time - Used to indicate the time when you want to start the schedule. On clicking Time Picker icon you can select the time.
Monthly	Used to set the value of frequency to monthly for the schedule.	Interval - Used to indicate the whether the schedule is scheduled for every month (a value of 1), every other month (a value of 2), or every third month (a value of 3) and so on. Month Day - Used to indicate the date of the month when you want to schedule the schedule. WeekDay - Used to indicate the day of the week when you want to schedule the schedule. Time - Used to indicate the time when you want to start the schedule. On clicking Time Picker icon you can select the time.
Weekly	Used to set the value of frequency to weekly for the schedule.	Interval - Used to indicate the whether the schedule is scheduled for every week (a value of 1), every other week (a value of 2), or every third week (a value of 3) and so on. WeekDay - Used to indicate the day of the week when you want to schedule the schedule. Time - Used to indicate the time when you want to start the schedule. On clicking Time Picker icon you can select the time.
Daily	Used to set the value of frequency to daily for the schedule.	Interval - Used to indicate the whether the schedule is scheduled for every day (a value of 1), every other day (a value of 2), or every third day (a value of 3) and so on.

Copyright © 2024, Oracle and/or its affiliates.

List Options	Description	Parameter Values
		WeekDay - Used to indicate the day of the week when you want to schedule the schedule. Time - Used to indicate the time when you want to start the schedule. On clicking Time Picker icon you can select the time.
Hourly	Used to set the value of frequency to hourly for the schedule.	Interval - Used to indicate the whether the schedule is scheduled for every hour (a value of 1), every other hour (a value of 2), or every third hour (a value of 3) and so on. WeekDay - Used to indicate the day of the week when you want to schedule the schedule. Time - Used to indicate the time in minute and second format when you want to start the schedule. On clicking Time Picker icon you can select the time.
Minutely	Used to set the value of frequency to minutely for the schedule.	Interval - Used to indicate the whether the schedule is scheduled for every minute (a value of 1), every other minute (a value of 2), or every third minute (a value of 3) and so on. WeekDay - Used to indicate the day of the week when you want to schedule the schedule.
Secondly	Used to set the value of frequency to secondly for the schedule.	Interval - Used to indicate the whether the schedule is scheduled for every second (a value of 1), every other second (a value of 2), or every third second (a value of 3) and so on. WeekDay - Used to indicate the day of the week when you want to schedule the schedule.

6. Click **Save**.
   The scheduler schedule is added.

Related Topics

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	See...
Search Scheduler Schedule zone	Search Scheduler Schedule on page 1076
How to search for a scheduler schedule	Searching for a Scheduler Schedule on page 1077
How to view the scheduler schedule details	Viewing a Scheduler Schedule Details on page 1081
How to edit a scheduler schedule	Editing a Scheduler Schedule on page 1081
How to delete a scheduler schedule	Deleting a Scheduler Schedule on page 1084

Viewing a Scheduler Schedule Details

Procedure

To view a scheduler schedule:

1. Search for the scheduler schedule in the **Search Scheduler schedule** zone of the **Scheduler Schedule** screen.

   Note: ORMB search engine supports wildcard search, where you can substitute the percentage (%) symbol as a stand in for any word or letter in a search criteria. You can use the '%' wildcard character in all input fields except the date and ID fields. The '%' wildcard character is suffixed automatically at the end of the partial search criteria. Therefore, you may or may not specify the wildcard character at the end of the partial search criteria. However, you have to prefix the wildcard character manually wherever required.

2. In the **Search Result** section, click the link in the **Schedule Name** column whose details you want to view.
   The **Scheduler Schedule** screen appears.
3. View the details of scheduler schedule in the **Scheduler Schedule** screen.

Related Topics

For more information on...	See...
Search Scheduler Schedule zone	Search Scheduler Schedule on page 1076
How to search for a scheduler schedule	Searching for a Scheduler Schedule on page 1077
How to define a scheduler schedule	Defining a Scheduler Schedule on page 1077
How to edit a scheduler schedule	Editing a Scheduler Schedule on page 1081
How to delete a scheduler schedule	Deleting a Scheduler Schedule on page 1084

Editing a Scheduler Schedule

Prerequisites

To edit a scheduler schedule, you should have:

- Schedule defined in the system.

Procedure

To edit a scheduler schedule:

1. Search for the scheduler schedule in the **Scheduler Schedule** screen whose details you want to edit.
2. In the **Search Results** section, click the **Edit** (✎) icon in the **Edit** column corresponding to the scheduler schedule whose details you want to edit.
   The **Scheduler Schedule** screen appears. It contains the following fields:

Copyright © 2024, Oracle and/or its affiliates.

Field Name	Field Description	Mandatory (Yes or No)
Schedule Name	Displays the name of scheduler schedule.	Not applicable
Detailed Description	Used to specify the description of the scheduler schedule.	No
Repeat Interval	Used to indicate the interval when the system should start the job scheduling.	Yes
Frequency	Used to indicate the frequency of the job recurrence. **Tip:** To select frequency type click the **Generate** button.	Yes

3. To set a time interval, click **Generate** button.
4. Select the frequency option from the drop-down list of **Frequency**.
   It contains the following options:

List Options	Description	Parameter Values
Yearly	Used to set the value of frequency to yearly for the schedule.	Interval - Used to indicate the whether the schedule is scheduled for every year (a value of 1), every other year (a value of 2), or every third year (a value of 3) and so on. Month and Day - Used to indicate the date and month when you want to schedule the schedule. Time - Used to indicate the time when you want to start the schedule. On clicking Time Picker icon you can select the time.
Monthly	Used to set the value of frequency to monthly for the schedule.	Interval - Used to indicate the whether the schedule is scheduled for every month (a value of 1), every other month (a value of 2), or every third month (a value of 3) and so on. Month Day - Used to indicate the date of the month when you want to schedule the schedule. WeekDay - Used to indicate the day of the week when you want to schedule the schedule. Time - Used to indicate the time when you want to start the schedule. On clicking Time

Copyright © 2024, Oracle and/or its affiliates.

List Options	Description	Parameter Values
		Picker icon you can select the time.
Weekly	Used to set the value of frequency to weekly for the schedule.	Interval - Used to indicate the whether the schedule is scheduled for every week (a value of 1), every other week (a value of 2), or every third week (a value of 3) and so on. WeekDay - Used to indicate the day of the week when you want to schedule the schedule. Time - Used to indicate the time when you want to start the schedule. On clicking Time Picker icon you can select the time.
Daily	Used to set the value of frequency to daily for the schedule.	Interval - Used to indicate the whether the schedule is scheduled for every day (a value of 1), every other day (a value of 2), or every third day (a value of 3) and so on. WeekDay - Used to indicate the day of the week when you want to schedule the schedule. Time - Used to indicate the time when you want to start the schedule. On clicking Time Picker icon you can select the time.
Hourly	Used to set the value of frequency to hourly for the schedule.	Interval - Used to indicate the whether the schedule is scheduled for every hour (a value of 1), every other hour (a value of 2), or every third hour (a value of 3) and so on. WeekDay - Used to indicate the day of the week when you want to schedule the schedule. Time - Used to indicate the time in minute and second format when you want to start the schedule. On clicking Time Picker icon you can select the time.
Minutely	Used to set the value of frequency to minutely for the schedule.	Interval - Used to indicate the whether the schedule is

Copyright © 2024, Oracle and/or its affiliates.

List Options	Description	Parameter Values
		scheduled for every minute (a value of 1), every other minute (a value of 2), or every third minute (a value of 3) and so on. WeekDay - Used to indicate the day of the week when you want to schedule the schedule.
Secondly	Used to set the value of frequency to secondly for the schedule.	Interval - Used to indicate the whether the schedule is scheduled for every second (a value of 1), every other second (a value of 2), or every third second (a value of 3) and so on. WeekDay - Used to indicate the day of the week when you want to schedule the schedule.

5. Click **Save**.
   The scheduler schedule is modified.

Related Topics

For more information on...	See...
Search Scheduler Schedule zone	Search Scheduler Schedule on page 1076
How to search for a scheduler schedule	Searching for a Scheduler Schedule on page 1077
How to define a scheduler schedule	Defining a Scheduler Schedule on page 1077
How to view the scheduler schedule details	Viewing a Scheduler Schedule Details on page 1081
How to delete a scheduler schedule	Deleting a Scheduler Schedule on page 1084

Deleting a Scheduler Schedule

Procedure

To delete a scheduler schedule:

1. Search for the scheduler schedule in the **Search Scheduler Schedule** zone of the **Scheduler Schedule** screen which you want to delete.

   Note: ORMB search engine supports wildcard search, where you can substitute the percentage (%) symbol as a stand in for any word or letter in a search criteria. You can use the '%' wildcard character in all input fields except the date and ID fields. The '%' wildcard character is suffixed automatically at the end of the partial search criteria. Therefore, you may or may not specify the wildcard character at the end of the partial search criteria. However, you have to prefix the wildcard character manually wherever required.

2. In the **Search Result** section, click the **Delete** (🗑) icon in the **Delete** column corresponding to the scheduler schedule which you want to delete.
   A message appears confirming whether you want to delete the scheduler schedule.
3. Click **OK**.
   The scheduler schedule is deleted.

Related Topics

Copyright © 2024, Oracle and/or its affiliates.

For more information on...	**See...**
Search Scheduler Schedule zone	Search Scheduler Schedule on page 1076
How to search for a scheduler schedule	Searching for a Scheduler Schedule on page 1077
How to define a scheduler schedule	Defining a Scheduler Schedule on page 1077
How to view the scheduler schedule details	Viewing a Scheduler Schedule Details on page 1081
How to edit a scheduler schedule	Editing a Scheduler Schedule on page 1081

Copyright © 2024, Oracle and/or its affiliates.

Appendix A

Administration

Topics:

- Feature Configurations

This section lists some of the administrative tasks.

Feature Configurations

This section explains how to set various feature configurations which are required for implementing various features in Oracle Revenue Management and Billing.

Setting the C1_CMO Feature Configuration

The **C1_CMO** feature configuration enables the system to determine the following:

- Administrative contracts and excess credit transactions which should be listed in the **Customer 360° Information** screen
- Payments against the suspense, on account, and excess credit contracts which are eligible for partial payment transfer from the payment list

Prerequisites

To set the **C1_CMO** feature configuration, you should have:

- Required contract types defined in the application

Procedure

To set the **C1_CMO** feature configuration:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **F** and then click **Feature Configuration**.
   A sub-menu appears.
3. Click the **Search** option from the **Feature Configuration** sub-menu.
   The **Feature Configuration Query** screen appears.
4. In the **Feature Configuration Search** zone, enter **C1_CMO** in the **Feature Name** field.

   Note: ORMB search engine supports wildcard search, where you can substitute the percentage (%) symbol as a stand in for any word or letter in a search criteria. You can use the '%' wildcard character in all input fields except the date and ID fields. The '%' wildcard character is suffixed automatically at the end of the partial search criteria. Therefore, you may or may not specify the wildcard character at the end of the partial search criteria. However, you have to prefix the wildcard character manually wherever required.
5. Click **Search**.
   A list of feature configurations that meet the search criteria appears in the search results.
6. In the **Search Results** section, click the link in the **Description** column corresponding to the feature configuration whose details you want to edit.

   The **Feature Configuration** screen appears. It contains the following option types:

Option Type	Description	Mandatory (Yes or No)
Administrative Contract Type	Used to specify the contract type. The contracts of the specified contract type are then listed in the **Administrative Contracts** zone of the **Account** tab on the **Customer 360° Information** screen.	No

Copyright © 2024, Oracle and/or its affiliates.

Option Type	Description	Mandatory (Yes or No)
	Note: If you want to specify multiple contract types, you need to set the **Administrative Contract Type** option type multiple times with a unique sequence number. If you do not specify the value for this option type, the data does not appear in the **Administrative Contracts** zone.	
On Account Contract Type	Used to specify the contract type. The excess credit transactions on the contracts of the specified contract type are then listed in the **Billed Vs Paid** zone of the **Account** tab on the **Customer 360° Information** screen. It also helps the system to determine the payments against the on account contracts (created using the specified contract type) which are eligible for partial payment transfer from the payment list. **Note:** If you want to specify multiple contract types, you need to set the **On Account Contract Type** option type multiple times with a unique sequence number. If you do not specify the value for this option type, the data does not appear in the **Billed Vs Paid** zone.	No
Excess Credit Contract Type	Used to specify the contract type. It helps the system to determine the payments against the excess credit contracts (created using the specified contract type) which are eligible for partial payment transfer from the payment list. Ideally, these excess credit contracts of the accounts should be used to park overpayments on the bills. **Note:** If you want to specify multiple contract types, you need to set the **Excess Credit Contract Type** option type multiple times with a unique sequence number.	No
Suspense Contract Type	Used to specify the contract type. It helps the system to determine the payments against the suspense contracts (created using the specified contract type) which are eligible for partial payment transfer from the payment list. **Note:** If you want to specify multiple contract types, you need to set the **Suspense Contract Type** option type multiple times with a unique sequence number.	No
Excluded Contract types for netting	Used when you want to exclude unmatched financial transactions of a particular contract type while creating	Yes (Conditional)

Copyright © 2024, Oracle and/or its affiliates.

Option Type	Description	Mandatory (Yes or No)
	transfer adjustments for a refund/write off request at the account level.	**Note:** This data is required when the adjustment level in the refund/write off request is set to **Account** or **Bill**.
	Note: You can exclude multiple contract types by defining the **Excluded Contract types for netting** option type multiple times (with unique sequence number) in the feature configuration.	

7. Enter the values for the required option types in the **Feature Configuration** screen.
8. Click the **Save** button in the **Page Title** area.
   The changes made to the feature configuration are saved.

Related Topics

For more information on...	See...
Administrative Contracts zone	Administrative Contracts
Billed Vs Paid zone	Billed Vs Paid
Partial Payment Transfer	Partial Payment Transfer on page 316

Setting the C1_FINTRANOP Feature Configuration

The **C1_FINTRANOP** feature configuration enables the system to determine various options which should be considered while creating billable charges, determining a price item pricing, generating bill segments, and while creating an actual bill using a trial bill.

Prerequisites

To set the **C1_FINTRANOP** feature configuration, you should have:

- Values defined for the **BIRC_OPT_TYP_FLG** lookup field

Procedure

To set the **C1_FINTRANOP** feature configuration:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **F** and then click **Feature Configuration**.
   A sub-menu appears.
3. Click the **Search** option from the **Feature Configuration** sub-menu.
   The **Feature Configuration Query** screen appears.
4. In the **Feature Configuration Search** zone, enter **C1_FINTRANOP** in the **Feature Name** field.

   Note: ORMB search engine supports wildcard search, where you can substitute the percentage (%) symbol as a stand in for any word or letter in a search criteria. You can use the '%' wildcard character in all input fields except the date and ID fields. The '%' wildcard character is suffixed automatically at the end of the partial search criteria. Therefore, you may or may not specify the wildcard character at the end of the partial search criteria. However, you have to prefix the wildcard character manually wherever required.

5. Click **Search**.
   A list of feature configurations that meet the search criteria appears in the search results.
6. In the **Search Results** section, click the link in the **Description** column corresponding to the feature configuration whose details you want to edit.

Copyright © 2024, Oracle and/or its affiliates.

The **Feature Configuration** screen appears. It contains the following option types:

Option Type	Description	Mandatory (Yes or No)
Account Financial History Details Switch	Used to indicate whether detailed or summarized information should be displayed for an account in the **Account Financial History** screen. The valid values are: • **Y** - Used when you want to list all financial transactions of a bill, payment, or adjustment for an account in the **Account Financial History** screen. • **N** - Used when you want to summarize all financial transactions of a particular type on an entity (i.e. bill, payment, or adjustment) and display the total amount in the **Account Financial History** screen. **Note:** If you do not specify the value for this option type, by default, it is set to **Y**.	No
Allow Advance Billing of Future Dated Charges	Used to indicate whether you want to use advance billing for future dated billable charges. The valid values are: • **Y** • **N** **Note:** If the **Allow Advance Billing of Future Dated Charges** option type is set to **Y**, the system considers those billable charges where the bill after date is on or before the cut-off date, irrespective whether the date of those billable charges is past or future dated. The **Allow Advance Billing of Future Dated Charges** option type should be defined only in case of non-recurring billable charges. By default, the value for this option type is set to **N**.	No
Allow Caching Of Rule Based Autopay List	Used to indicate whether you want to cache the rule based auto pay instructions of an account when the **C1-APRULEEVL** algorithm is invoked. The valid values are: • **Y** • **N** **Note:** If you do not specify the value for this option type, by default, it is set to **N**.	No
Allow non-significant zero amount FT GL entry	Used to indicate whether you want to allow non-significant zero amount **FT GL** entries in the system. The valid values are:	Yes (Conditional)

Copyright © 2024, Oracle and/or its affiliates.

Option Type	Description	Mandatory (Yes or No)
	• **Y** • **N** **Note:** The **FT GL** amount is considered significant only when the statistical and total amount are non-zero.	**Note:** This data is required when you want to allow non-significant zero amount **FT GL** entries in the system.
Avoid internal match event creation for zero dollar Bill	Used to indicate whether you want to consider a zero balance bill for offsetting against another bills in an offset request. The valid values are: • **Y** • **N** **Note:** If you set this option type to **Y**, the system does not match the credit and debit line items internally of a zero balance bill on the bill completion. If you set this option type to **N**, the system matches the credit and debit line items internally of a zero balance bill on the bill completion.	Yes (Conditional) **Note:** This data is required when you want to consider a zero balance bill for offsetting against another bills in an offset request.
Avoid Pricing Crawling if Rate Schedule present at Contract	Used to indicate that the system should not determine the price item pricing using the price assignment search algorithm (which is attached on the account's division) when the effective rate schedule is present on the contract. The valid values are: • **Y** • **N** **Note:** If you do not specify the value for this option type, by default, it is set to **N**.	No
Bill Segment Regeneration on Freeze	Used to indicate whether you want to regenerate the bill segments before freezing them. The valid values are: • **Y** • **N** **Note:** If you do not specify the value for this option type, by default, it is set to **N**. If the **Freeze and Complete** option is selected for the respective window period in the bill cycle and the **Bill Segment Regeneration on Freeze** option type is set to **Y**, then the bill segments are regenerated and the status of the bill segments is set to **Frozen**. However, if the **Freeze and Complete** option is not selected for the respective window period in the bill cycle, then the bill segments are always regenerated irrespective of whether the **Bill Segment Regeneration on Freeze** option type is set to **Y** or **N**.	No

Copyright © 2024, Oracle and/or its affiliates.

Option Type	Description	Mandatory (Yes or No)
Billable Charges Date - Non Recurring	Used to indicate whether you want to consider the non-recurring billable charges whose start or end date is earlier than or equal to the cut-off date. The valid values are: • **S** - Used when you want to consider the non-recurring billable charges whose start date is earlier than or equal to the cut-off date. • **E** - Used when you want to consider the non-recurring billable charges whose end date is earlier than or equal to the cut-off date. **Note:** If you do not specify the value for this option type, by default, it is set to **S**. The system always considers the recurring billable charges whose start date is earlier than the cut-off date and end date is later than the cut-off date irrespective of whether the **Billable Charges Date** option type is set to **S** or **E**.	No
Billable Charges Date - Recurring	Used to indicate whether you want to consider the recurring billable charges whose start or end date is earlier than or equal to the cut-off date. The valid values are: • **S** - Used when you want to consider the recurring billable charges whose start date is earlier than or equal to the cut-off date. • **E** - Used when you want to consider the recurring billable charges whose end date is earlier than or equal to the cut-off date. **Note:** If you do not specify the value for this option type, by default, it is set to **S**.	No
Catch Up Flag For Recurring Charges	Used to indicate whether you want to consider the recurring billable charges for catch up. The valid values are: • **Y** - Used when you want to consider the recurring billable charges for catch up during the bill generation process, where the billable charge end date is earlier than the billable charge cut-off date. • **N** - Used when you do not want to consider the recurring billable charges for catch up during the bill generation process, where the billable charge end date is earlier than the billable charge cut-off date.	Yes (Conditional) **Note:** This data is required when you want to catch up with the recurring billable charges before the bill generation process.
Defer Bill Segment Freeze	Used to indicate whether you want to defer the freezing process of bill segments until the bill routing and due date calculation is completed. The valid values are:	Yes (Conditional)

Copyright © 2024, Oracle and/or its affiliates.

Option Type	Description	Mandatory (Yes or No)
	• **Y** - Used when you want the system to freeze the bill segment after the bill routing and bill due date calculation is completed. • **N** - Used when you do not want the system to freeze the bill segment after the bill routing and bill due date calculation is completed. **Note:** This flag is only considered for freezing the bill once the bill's status is set to **Complete**.	**Note:** This data is required when you want to defer the freezing process of bill segments until the bill routing and due date calculation is completed.
Hide Debt Class Details On Account Main Page	Used to indicate whether you want to hide the debt class details in the **Main** tab of the **Account** screen. The valid values are: • **Y** - Used when you want the system to hide the debt class details in the **Main** tab of the **Account** screen. • **N** - Used when you want the system to populate the debt class details in the **Main** tab of the **Account** screen.	Yes (Conditional) **Note:** This data is required when you want to hide the debt class details in the **Main** tab of the **Account** screen.
Rate Component Step Optimization	Used to indicate whether you want to determine the rate for all tiering ranges of a step tiered rate component or for only those tiering ranges of a step tiered rate component till when the quantity is exhausted. The valid values are: • **Y** - Used when you want to determine the rate for only those tiering ranges of a step tiered rate component till when the quantity is exhausted. It helps to improve the performance of the billing batches. If you set this option type to **Y**, always ensure that you define the price components with the tiering ranges in the ascending order. • **N** - Used when you want to determine the rate for all tiering ranges of a step tiered rate component. **Note:** If you do not specify the value for this option type, by default, it is set to **N**.	No
Refund/Write-Off Flush Counter	Used to specify the value for flush counter when the refund or write-off details are defined. **Note:** You must specify a numeric value greater than zero. If the total number of refund or write-off records are equal to the flush counter, the system then performs a flush and clear at session level.	Yes (Conditional) **Note:** This data is required when you are defining the refund or write-off details in the system.
Skip Priceitem Contract Type Validation	Used to indicate whether you want to skip the price item's and billable charge's contract's contract type validation. The valid values are: • **Y** - Used when you want to create a billable charge for a price item on the contract which is created using a contract type to which the price item is not associated.	No

Copyright © 2024, Oracle and/or its affiliates.

Option Type	Description	Mandatory (Yes or No)
	• **N** - Used when you want to create a billable charge for a price item on the contract which is created using a contract type to which the price item is associated. **Note:** If you do not specify the value for this option type, by default, it is set to **N**.	
Trial Bill Characteristics Conversion	Used to indicate whether you want to copy the trial bill's characteristics when an actual bill is created using a trial bill. The valid values are: • **Y** • **N** **Note:** If you do not specify the value for this option type, by default, it is set to **N**.	No

7. Enter the values for the required option types in the **Feature Configuration** screen.
8. Click the **Save** button in the **Page Title** area.
   The changes made to the feature configuration are saved.

Setting the C1_PERACCT Feature Configuration

The **C1_PERACCT** feature configuration enables the system to determine whether the **Add Account and Start Service** option should be provided while creating or editing a person. Accordingly, the **Add Account and Start Service** check box and the **Customer Class** field appear in the **Main** tab of the **Person** screen when you create, edit, or view the details of a person.

Procedure

To set the **C1_PERACCT** feature configuration:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **F** and then click **Feature Configuration**.
   A sub-menu appears.
3. Click the **Search** option from the **Feature Configuration** sub-menu.
   The **Feature Configuration Query** screen appears.
4. In the **Feature Configuration Search** zone, enter **C1_PERACCT** in the **Feature Name** field.

   Note: ORMB search engine supports wildcard search, where you can substitute the percentage (%) symbol as a stand in for any word or letter in a search criteria. You can use the '%' wildcard character in all input fields except the date and ID fields. The '%' wildcard character is suffixed automatically at the end of the partial search criteria. Therefore, you may or may not specify the wildcard character at the end of the partial search criteria. However, you have to prefix the wildcard character manually wherever required.

5. Click **Search**.
   A list of feature configurations that meet the search criteria appears in the search results.
6. In the **Search Results** section, click the link in the **Description** column corresponding to the feature configuration whose details you want to edit.

   The **Feature Configuration** screen appears. It contains the following option type:

Copyright © 2024, Oracle and/or its affiliates.

Option Type	Description	Mandatory (Yes or No)
Create Person Account Together	Used to indicate whether the **Add Account and Start Service** check box should appear in the **Main** tab of the **Person** screen. The valid values are: • Y • N **Note:** On clicking the **Add Account and Start Service** option, you can create a new account for the person when you save the person information. The system then links the person to the account using the relationship type which is specified in the installation options.	Yes

7. Enter the value for the required option type in the **Feature Configuration** screen.
8. Click the **Save** button in the **Page Title** area.
   The changes made to the feature configuration are saved.

Related Topics

For more information on...	See...
Person screen	Person - Main on page 4

Setting the C1-RECLMT Feature Configuration

The **C1-RECLMT** feature configuration enables you to configure the maximum number of records that can be retrieved through the following business services:

- C1-GetAdjustmentHistory
- C1-ViewPaymentService

If you specify a very high record limit, the performance of the business services would be impacted. Therefore, we recommend you to specify a value which meets the performance requirements.

Procedure

To set the **C1-RECLMT** feature configuration:

1. Click the **Admin** link in the **Application** toolbar.
   A list appears.
2. From the **Admin** menu, select **F** and then click **Feature Configuration**.
   A sub-menu appears.
3. Click the **Search** option from the **Feature Configuration** sub-menu.
   The **Feature Configuration Query** screen appears.
4. In the **Feature Configuration Search** zone, enter **C1-RECLMT** in the **Feature Name** field.

   Note: ORMB search engine supports wildcard search, where you can substitute the percentage (%) symbol as a stand in for any word or letter in a search criteria. You can use the '%' wildcard character in all input fields except the date and ID fields. The '%' wildcard character is suffixed automatically at the end of the partial search criteria. Therefore, you may or may not specify the wildcard character at the end of the partial search criteria. However, you have to prefix the wildcard character manually wherever required.

5. Click **Search**.
   A list of feature configurations that meet the search criteria appears in the search results.

Copyright © 2024, Oracle and/or its affiliates.

6. In the **Search Results** section, click the link in the **Description** column corresponding to the feature configuration whose details you want to edit.

   The **Feature Configuration** screen appears. It contains the following option type:

Option Type	Description	Mandatory (Yes or No)
Set Limit for IWS Retrieving Records	Used to indicate the maximum number of records that you can retrieve through the **C1-GetAdjustmentHistory** and **C1-ViewPaymentService** business services. **Note:** You must specify an integer value for this option type. If you do not specify the value for this option type, by default, it is set to **50**.	No

7. Enter the value for the required option type in the **Feature Configuration** screen.
8. Click the **Save** button in the **Page Title** area.
   The changes made to the feature configuration are saved.

Related Topics

For more information on...	See...
C1-GetAdjustmentHistory business service	C1-GetAdjustmentHistory Business Service on page 506
C1-ViewPaymentService business service	C1-ViewPayment Business Service on page 443

Copyright © 2024, Oracle and/or its affiliates.