

**Oracle Financial Services Revenue
Management and Billing**

OR

**Oracle Insurance Revenue
Management and Billing**

Version 6.1.0.0.0

Release Notes

Revision 1.1

F90871-01

January 2024

Oracle Financial Services Revenue Management and Billing/Oracle Insurance Revenue Management and Billing Version 6.1.0.0.0 Release Notes

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

F90871-01

Document Category: Public

Copyright Notice

Copyright © 2009, 2024, 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.

About this Document

This document describes the new features, enhancements, user interface and database level changes, supported platforms, framework upgrade, supported upgrades, and technology upgrade made in this release. It also highlights the discontinued features, bug fixes, and known issues in this release.

This document does not describe the functionality of Oracle Revenue Management and Billing or technical know-how on how to install or upgrade Oracle Revenue Management and Billing. For more detailed information, you can refer to the following documents:

Document Name	Description
<i>Oracle Revenue Management and Billing Licensing Guide</i>	<p>Lists different features which you can use when you acquire a license for the following products:</p> <ul style="list-style-type: none"> • Oracle Financial Services Revenue Management and Billing • Oracle Insurance Revenue Management and Billing <p>It also provides the licensing information about the third-party JARs and components which are included in the above-mentioned products.</p>
<i>Oracle Revenue Management and Billing Quick Installation Guide</i>	<p>Provides information about the media packs which are available for the current release. It contains information about the supported platforms, space requirements, and release-specific documentation library. In addition, it contains high-level information on how to install ORMB and selected additional software.</p>
<i>Oracle Revenue Management and Billing Installation Guide</i>	<p>Provides an overview about the application architecture. It contains detailed information about the software and hardware requirements, supported platforms, application server and database space requirements, and application server pre-requisites for supported platforms. It explains the installation and configuration menu options which are available during installation and advanced menu options which are available post installation. It also explains how to configure WebSphere application server and how to install OUAF, OUAF rollup packs, ORMB, and ORMB patches or rollup packs.</p> <p>It contains appendices which lists ORMB rollup packs and notices about third-party JAR and other components.</p>

Document Name	Description
<i>Oracle Revenue Management and Billing Database Administrator's Guide</i>	<p>Provides information about the supported database platforms and explains how to install database with or without demo data. It contains the standards and conventions that should be followed while working with ORMB database. In addition, it lists various configurations and implementation guidelines with respect to tablespace, encryption, storage, compression, indexes, initialization parameters, listener, table partitioning, performance monitoring, memory resource management, optimizer statistics and segment advisors, etc.</p> <p>It contains various appendices which lists the new objects added in ORMB, application services configured for default user groups, new objects added in OUAF, information about OUAF system tables and ORMB rollup packs, notices about third-party JAR and other components.</p>
<i>Oracle Revenue Management and Billing Security Guide</i>	<p>Highlights what's new in security! Describes all the security features available in ORMB for authentication, authorization, user access, database access, LDAP and SSO integration, audit, encryption, web services, and JNDI access. It also describes how to provide support for data masking, Oracle cloud object storage, groovy, HTTP proxy, keystore, truststore, whitelist, federated architecture, and object erasure. In addition, it explains how to garble the customer information.</p>
<i>Oracle Revenue Management and Billing Server Administration Guide</i>	<p>Provides detailed information about the product architecture, native support, directory structures, environment variables, logs, etc. It explains various concepts related to the batch server and lists and describes the scripts required for installing and configuring ORMB. It explains how to monitor the system and configure Web browser, Web application server, business application server, inbound web services, and batch server. It explains how to provide Oracle Cloud Support and how to integrate and monitor Oracle Scheduler.</p> <p>It contains various appendices which lists and describes the parameters available in the ENVIRON.INI, Web.xml, spl.properties, hibernate.properties, submitbatch.properties, threadpoolworker, coherence-cache.config.xml, and tangosol-coherence-override.xml files.</p>

Document Name	Description
<i>Oracle Utilities Application Framework Business Process Guide</i>	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.
<i>Oracle Utilities Application Framework Administrative Guide</i>	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 on 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).
<i>Oracle Revenue Management and Billing Business Process Guide</i>	<p>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.</p> <p>The features listed and described in this document can be used in both financial services and health insurance domains.</p>
<i>Oracle Revenue Management and Billing Administrative Guide</i>	<p>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.</p> <p>The information available in this document can be used in both financial services and health insurance domains.</p>

Document Name	Description
<i>Oracle Revenue Management and Billing Banking User Guide</i>	<p>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 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 management, etc.</p> <p>It describes all screens related to these features and explains how to perform various tasks related to the feature in the application.</p>
<i>Oracle Revenue Management and Billing Insurance User Guide</i>	<p>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.</p> <p>It describes all screens related to these features and explains how to perform various tasks related to the feature in the application.</p>
<i>Oracle Revenue Management and Billing Upgrade Path Guide</i>	<p>Explains the path and pre-requisites for upgrading Oracle Revenue Management and Billing from one version to another.</p>

Document Name	Description
<i>Oracle Revenue Management and Billing Upgrade Guide</i>	<p>Explains how to upgrade the ORMB application server and database from one version to another. It also explains how to migrate the ORMB data from one version to another and describes the additional tasks that you need to perform after upgrading from one version to another.</p> <p>It includes various appendices that contain information about new tables introduced in the current release, existing tables which are modified in the current release, dropped algorithms and algorithm types, dropped characteristic types, dropped algorithm parameters, dropped option types in feature configurations, ORMB rollup packs, and SQL statements used for data migration.</p>
<i>Oracle Revenue Management and Billing Direct Database Upgrade Guide</i>	<p>Explains how to directly upgrade the ORMB database from 2.5.0.1.0 or any later version to the current release. It also highlights any known issues during direct database upgrade and how to handle these issues in the database.</p>
<i>Oracle Revenue Management and Billing Transaction Feed Management - Batch Execution Guide</i>	<p>Explains the sequence in which the batches should be executed while performing various tasks in the Transaction Feed Management (TFM) module.</p> <p>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.</p>
<i>Oracle Revenue Management and Billing Batch Guide</i>	<p>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.</p>
<i>Oracle Revenue Management and Billing Information Lifecycle Management (ILM) Implementation Guide</i>	<p>Provides an overview of the Information Lifecycle Management (ILM) feature. It describes how to implement ILM for the Transaction Feed Management (TFM) and Billing modules. It also provides detailed information about the ILM batches and their parameters.</p>
<i>Oracle Revenue Management and Billing FOP Reports Guide</i>	<p>Explains how to extract data from the system using various FOP reports in Oracle Revenue Management and Billing.</p>
<i>Oracle Revenue Management and Billing Chatbot Configuration Guide</i>	<p>Explains how to integrate Oracle Digital Assistant (ODA) with Oracle Revenue Management and Billing.</p>

Document Name	Description
<i>Oracle Revenue Management and Billing Chatbot User Guide</i>	Explains how to use the menu based Chatbot introduced in Oracle Revenue Management and Billing.
<i>Oracle Revenue Management and Billing ML Integration Guide</i>	Explains how to integrate Machine Learning (ML) with Oracle Revenue Management and Billing for anomaly detection.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Change Log

Revision	Last Update	Updated Section	Comments
1.1	12-Apr-2024	Delinquency Management	Updated Information

Contents

Prerequisites	1
New Features (Generic).....	2
Division Group.....	2
User Status Synchronization (GBUCS and CFS Environments).....	2
New Features (Specific to Financial Services)	4
Unified Search Feature.....	4
Characteristic Support for Deal Price List Assignment.....	5
Standard Pricing Comparison in Deal through Inbound Web Service	6
Adjust Underpayments at Bill Level through Write Off	7
Default Bill Cutoff Date	10
New Features (Specific to Insurance)	2
Automatic Offset of Debit and Credit Bill Line Items.....	2
To Do Notification on Binder Payment Cancellation	7
Transfer Binder Payment from General Suspense Account to Individual Membership Account.....	9
Reconciliation of Individual Health Insurance Subsidy Payments	13
Reconciliation Discrepancy Report of Individual Health Insurance Subsidy Payments	17
Deferred Processing Mode for Discrepancy Report Creation.....	18
Invoice Request for Individual Health Insurance Billing	19
Manual Regular Bill Generation Invoice Request	22
Manual Trial Bill Generation Invoice Request.....	23
Automatic Regular Bill Generation Invoice Request.....	23
Collection Class	27
Delinquency Management.....	29
Group Health Insurance Delinquency	31
Individual Health Insurance Delinquency	32
Account Level Delinquency Process.....	32
Account Level Delinquency Process Creation	33
Person Level Delinquency Process.....	37
Person Level Delinquency Process Creation	38
Types of Delinquency Processes	42
Debt Notification Delinquency Process	43

Debt Collection Letters Delinquency Process	45
Individual Membership Cancellation Delinquency Process	47
Individual Membership Termination Delinquency Process	50
Advance Deposit Delinquency Process	55
Delinquency Control	56
Delinquency Event Type.....	61
Delinquency Event Activation	63
Delinquency Event Completion.....	81
Delinquency Process Type	82
Delinquency Process Type Details	83
Delinquency Event Trigger Date Calculation Details.....	84
Delinquency Event Types	90
Delinquency Process Type Algorithms.....	91
Grace Period Configuration	109
Termination and Reinstatement Configuration.....	110
Termination Date Rule Priority	111
Trigger Date Recalculation Configuration.....	111
Approval Configuration.....	112
Delinquency Event Eligibility Evaluation and Trigger Date Calculation	113
Delinquency Process Maintenance.....	115
Manual Actions at the Delinquency Process Level	115
Manual Actions at the Delinquency Event Level	118
Delinquency Process Approval.....	121
Canceling a Delinquency Process.....	122
Resuming a Delinquency Process.....	124
Resuming a Delinquency Process on Payment Cancellation.....	124
Resuming a Delinquency Process on Adjustment Cancellation.....	127
Recalculating Trigger Dates for Pending Events	130
Membership Cancellation Outbound Message	130
Membership Termination Outbound Message	131
Creating Customer Contacts for Group or Individual Memberships	132
DELINPROC Feature Configuration	135
Outbound Message Generation for Delinquency Management	135
Delinquency Grace Period Business Rule.....	137
Delinquency Event Attribute Business Rule.....	138

Delinquency Termination Date Rule Business Rule	139
Delinquency Miscellaneous Options Business Rule	142
Hold Repricing	145
Maintaining Address Status	146
Prorate Membership Benefit and Corresponding Sponsor Charges.....	148
Zero-Amount Membership Benefit Charges	149
Additional Repricing Entity Detail Records	152
Allocate Advance Deposit for Delinquent Customers.....	153
Maintaining Advance Deposit Details	154
Applying Advance Deposit Payments	155
Monitoring Advance Deposit of a Parent Customer	156
Settling Unpaid Dues Against Advance Deposit through a Delinquency Process.....	158
Notification on Receiving Advance Deposit.....	162
Notification on Advance Deposit Cancellation	163
Accounts Payable (Refund Adjustments) Extraction	165
Updating Accounts Payable Request in ORMB	166
Delinquency Process Type Preference	168
Delinquency Process Preference.....	170
Routing Method – Contact Method Mapping Preference.....	173
Enhancements (Generic)	174
C1-FLUPLD Feature Configuration	174
File Upload Interface.....	174
FOP Reporting	174
Enhancements (Specific to Financial Services)	176
Billing Anomaly Detection with Artificial Intelligence (AI) and Machine Learning (ML).....	176
Statements in TWIST and CAMT Formats	179
Enhancements (Specific to Insurance).....	180
Inbound Message.....	181
Offset Request	182
Refund/Write Off Request	183
Payment Request.....	183
Customer 360° View.....	184
Binder Payment Preference (Field Mapping).....	184
Health Product 360° View	184
Reconciliation.....	185

Membership (Screen).....	186
Invoice Request.....	186
Individual Health Insurance Billing.....	187
Individual Health Insurance Pricing.....	187
Field Mapping	188
Business Rules	188
Parameter	188
Collection Class	189
Customer Class.....	189
Adjustments	189
C1-MembershipRepricing Business Service	190
Payments.....	190
Individual Membership	191
Subscription Tier Derivation for Membership	193
Repricing	194
C1-ASOBLLNG Feature Configuration	194
C1_CMO Feature Configuration.....	196
C1-INVREQ Feature Configuration.....	196
Membership Benefits.....	196
Fully Insured Pricing.....	196
Fully Insured Group Policy	197
Framework Upgrade	198
Redwood User Experience	198
Menu Item Search.....	198
Upgrade Impact.....	198
User Interface (UI) Level Changes	200
Database Level Changes	208
New Objects in the ORMB V6.1.0.0.0 Database	208
New Objects in the OUAF V4.5.0.1.1 Database	208
New Tables in the ORMB V6.1.0.0.0 Database	208
New Tables in the OUAF V4.5.0.1.1 Database	209
Existing Tables Modified in ORMB V6.1.0.0.0.....	209
Existing Tables Modified in OUAF V4.5.0.1.1.....	209
Algorithms and Algorithm Types Dropped in ORMB V6.1.0.0.0.....	209
Algorithms and Algorithm Types Dropped in OUAF V4.5.0.1.1	209

Parameters Added or Removed from Algorithm Types in ORMB V6.1.0.0.0	209
Option Types Added or Removed from Feature Configurations in ORMB V6.1.0.0.0.....	210
Characteristic Types Dropped in ORMB V6.1.0.0.0	210
Batch Controls Dropped in ORMB V6.1.0.0.0	210
Parameters Added or Removed from Batch Controls in ORMB V6.1.0.0.0	210
Parameters Added or Removed from Batch Controls in OUAF V4.5.0.1.1	210
Default User Group Application Services.....	211
Supported Platforms	212
Technical Recommendations	214
Supported Upgrades	215
Unsupported Integrations	216
Oracle Utilities Application Framework Version 4.5.0.1.1 Enhancements.....	217
Application Security Enhancements	217
Detailed Description Added to Application Service.....	217
Screen Information Hidden on Session Timeout	218
Product Usability Enhancements	218
Algorithm Portal	219
Application Security Query Portal	219
Application Service Query Portal - Search by Release Version.....	219
Characteristic Type Portal - Support Large Number of Characteristic Values	219
Currency External Reference.....	220
Currency Portal	220
Dashboard Location Values Adjustment	220
Display Icon Portal.....	221
Geographic Type Portal.....	222
Improved Field Portal.....	222
Inbound Web Service Query Portal - Additional Search Options	222
Language Portal	222
Lookup Portal	222
Menu Portal	223
Online Help Opens in Oracle Help Center for Oracle Utilities Application Framework Based Applications.....	223
Phone Type Portal	223
Shortcut Key for Navigating Multiple Tabs Adjusted.....	223
Unified Search Improvements	224

User Group Portal	224
Work List Zone - Improved Item Navigation	225
Data Export Enhancements.....	226
Data Redaction Rules	226
General Data Export - Restrict Initial Export by Time.....	227
Generalized Export Dashboard Portal Enhancements.....	227
Generalized Data Export Support for Entity Log Records	228
Generalized Data Export Support for Lost Ongoing Export Files.....	228
To Do Management and Processing Enhancements.....	228
To Do Management by Assigned User	228
To Do Dashboard Details Tab	228
Batch Processing Enhancements	229
Batch Logs for Issues Found Before Thread Execution	229
Batch Submission Parameters Defaulted from Batch Control	229
Manifest File Includes Previous Manifest File Information.....	230
Plug-in Driven Batch Variables	230
Plug-in-driven Extract - Support for Writing to Multiple Files and Flexibility in File Names	231
Submit Batch Job REST Linux Script.....	231
Support for Encrypted Files and Digital Signatures.....	232
Support Override of Base Algorithm on Batch Control.....	232
Implementation Tool Enhancements.....	233
Action Provided to MO Audit Plug-In Spot.....	233
Attachment Malware Scan Plug-in.....	233
Base Product Index Data Visible in Table Portal.....	234
Business Object Status Reasons REST API	234
Characteristic Mapping Language Genericized.....	234
Debugging Tools - Consolidated into a Single Button	234
Dropdown Lists REST API.....	235
Expand Tree REST API.....	235
Extensions Dashboard Portal	235
HTML Editor Syntax Highlighted	236
HTML Row Header Reference Update	236
Insights REST API	236
Javadocs Viewer	237
Menu Item Configuration for Add Action.....	237

New Base Display Icon Images	237
Parameter Update for New Language Batch Program.....	238
Content Migration Assistant (CMA) Enhancements	239
CMA Import Performance Improvements.....	239
CMA Support for Separate Configuration and Business Data Migrations	240
Export Content Migration Assistant Data to an Older Version.....	240
Migration Plan Pre-Compare Algorithm Execution Update	241
Miscellaneous Enhancements.....	242
Application Service's Application Security Zones	242
Cloud Object Storage Support for Platform as a Service (PaaS)	243
Digital Self Service Masquerading Using Key Ring	243
HTML Sanitization Improvements.....	243
Improved Characteristic Type Referential Integrity	244
Key Ring Support for OAuth Client and Secret.....	245
New Class Element on Business Flag Standard Name BusinessObject.....	245
Support for Capturing and Exporting Batch Run Analytics Data	245
Deprecation Notices for OUA Version 4.5.0.1.1	247
Deprecation in This Release	247
Support for Migration Requests F1-FrameworkAdmin and F1-SchemaAdmin	247
Work Calendar Legacy Page Metadata	248
Time Zone Legacy Page Metadata	248
Application Viewer	248
Invite User to Mobile Application Zone	248
Mobile Remote Message Artifacts.....	248
Mobile Data Terminal Artifacts	248
Mobile Component Artifacts.....	248
Mobile Data Terminal Type Artifacts.....	249
Deployment Part Artifacts.....	249
Deployment Type Artifacts.....	249
Deployment Artifacts	249
Master Configuration Business Objects Related to Mobile ApplicationFramework.....	249
Migration Plan - Import Algorithms	249
User Group Service Management Portal	249
Deprecation Planned for Future Releases	250
Support for Previous User Experience	250

F1-DFLTAPS and FWLZDEXP Application Services	250
Workflow and Notification Metadata and Database Tables	251
Mobile Application Framework Metadata	251
Key Ring Validation Scripts, Algorithm Types, and Algorithms.....	251
UI Metadata Related to Converted Pages.....	252
Miscellaneous System Data	253
XSLT Managed Content Type	253
REST IWS - Original REST Servlet.....	253
Append Setting from Pagination.....	254
Support for Master/Subordinate Servers for Web Service Catalog.....	254
Batch Run Statistics Portal Functionality	254
Configuration Migration Assistant Import Algorithm Plugin Spot	254
F1-MAINPROC Business Object Read When Pre-processing Exists	254
Deprecation Notices for ORMB Version 6.1.0.0.0	255
Deprecation in This Release	255
Support for @INSATLL_DIR.....	255
Deprecated Platforms	255
Deprecation Planned for Future Releases	255
Product Documentation	257
Documentation Updates	258
Media Pack Download	259
Bug Fixes	260
Known Issues	266
Framework	266
Banking.....	267
Insurance.....	288
Documentation	291
Technical Support	293

Prerequisites

If a customer is already using the Transaction Feed Management feature and wants to upgrade to Oracle Revenue Management and Billing Version 6.1.0.0.0, then the customer needs to ensure the following (before upgrading):

- All bills generated in the system are in the **Complete** status. In other words, there should not be any bills in the **Pending** status. If there are any bills in the **Pending** status or if any billable charge (generated through TFM) is not yet billed, disaggregation and cancellation of transactions which are uploaded using any previous version will not happen successfully.
- Transactions which are uploaded using any previous version must not be in the **Initial Product Determined (INPD)** status. They can be in the **Uploaded (UPLD)**, **Invalid (INVL)**, **Error (EROR)**, **Completed (COMP)**, or **Cancelled (CNCL)** status.
- Equal to (=) or tilde (~) symbol is not used in any existing price item parameter code and value. Otherwise, erroneous results might occur.

New Features (Generic)

This section describes the following new features added in this release which can be used in both financial services and health insurance domains:

- [Division Group](#)
- [User Status Synchronization \(GBUCS and CFS Environments\)](#)

Division Group

Oracle Revenue Management and Billing enables you to group one or more divisions into a geographic region or segment. This grouping can be based on various factors, such as business cut-off time, source data file availability and volume, etc. You can then process the source data of a geographic region or segment via a single job schedule. This is done with the objective to support data processing from multiple divisions in the same batch job, and thereby achieve maximum utilization of resources and improved business performance. Note that each division in the geographic region or segment is a stand-alone entity and can be processed in any order.

To introduce this new feature, the following changes are made to the system:

- A new maintenance object named **C1-DIVGROUP** and a new business object named **C1-DivisionGroup** are introduced in this release.
- A new screen named **Division Group** is introduced in this release.

The **Division Group** screen is accessible from the **Admin** menu. It enables you to define a division group where you can include one or more divisions that are already defined in the system. It also enables you edit, delete, and copy a division group. You can set the status of a division group to **Active** or **Inactive**. The system will then allow you to use active division groups while executing batches in the system. Note that the **Division Group** parameter will be added in the required batches in the future release of ORMB. However, at present, if you want you can create the required division groups and add the **Division Group** parameter in the required custom batches.

User Status Synchronization (GBUCS and CFS Environments)

The **User Status Synchronization** feature is offered for Oracle Revenue Management and Billing Cloud Service and Oracle Revenue Management and Billing Cloud Service Premium Edition. There may be scenarios wherein the users are enabled or disabled in either Oracle Revenue Management and Billing (ORMB) or Oracle Identity Cloud Service (IDCS). Ideally, the user status should be in sync in both ORMB and IDCS. The system enables you to synchronize the user status from ORMB to IDCS and vice-versa. To synchronize the user status from ORMB to IDCS, you need to configure the **ORMB to IDCS User Sync** master configuration in ORMB. Here, in the master configuration, you need to specify the following:

- **IDCS User Sync Reqd** – Used to indicate whether you want to synchronize the user status from ORMB to IDCS.
- **IDCS URL** – Used to specify the authentication URL using which you access the ORMB Cloud Service environment. This field appears and is required when the **IDCS User Sync Reqd** option is selected.
- **Client ID** – Used to specify the public identifier for the ORMB Cloud Service environment. This field appears and is required when the **IDCS User Sync Reqd** option is selected.
- **Client Secret** – Used to specify a secret known only to the ORMB application and to the authorization server. This field appears and is required when the **IDCS User Sync Reqd** option is selected.
- **Application ID** – Used to specify the ORMB application that you want to access on Oracle Cloud Infrastructure (OCI). This field appears and is required when the **IDCS User Sync Reqd** option is selected.

After you setup the **ORMB to IDCS User Sync** master configuration, the system automatically updates IDCS immediately whenever you enable or disable the user in ORMB. To synchronize the user status from IDCS to ORMB, you need to execute a newly introduced batch named **C1-SYNUSB** in ORMB. This batch is used to update the user status to **Enable** or **Disable** in ORMB depending on whether the user is enabled or disabled in IDCS. It is a single-threaded batch. It contains the following parameter:

Parameter Name	Mandatory (Yes or No)	Description
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

New Features (Specific to Financial Services)

This section describes the following new features added in this release which can be used in the financial services domain:

- [Unified Search Feature](#)
- [Characteristic Support for Deal Price List Assignment](#)
- [Standard Pricing Comparison in Deal through Inbound Web Service](#)
- [Adjust Underpayments at Bill Level through Write Off](#)
- [Default Bill Cutoff Date](#)

Unified Search Feature

The **Search** field in the **Application** toolbar allows you to search for business entities (such as persons and accounts) in addition to the standard search for menu items. For example, you can search for a person using its name or primary identifier and account using its primary identifier. The unified search feature is a simplified version of search that allows you to search for records using free form search text or hint codes. For more complex queries, you may navigate to the corresponding advanced search screens.

You can use the following hints to search for the respective entities:

Hint	Description	Example
/	Used to search for a menu item to navigate directly to the corresponding page or BPA script rather than using the menus to navigate to the desired page or script.	/Customer Registration
n:	Used to search for a customer using the person or customer name. Here, you can specify the percentage (%) symbol as a wildcard character. The customers that meet the search criteria are shown in the drop-down list. Once you select a customer from the list, the Customer 360° Information screen appears with the details of the respective customer.	n: james n: leo%do
id:	Used to search for a customer using its primary identifier. Here, you cannot specify the percentage (%) symbol as a wildcard character. The customers that meet the search criteria are shown in the drop-down list. Once you select a customer from the list, the Customer 360° Information screen appears with the details of the respective customer.	id: 344234423

Hint	Description	Example
aid:	Used to search for an account using its primary identifier. Here, you cannot specify the percentage (%) symbol as a wildcard character. The accounts that meet the search criteria are shown in the drop-down list. Once you select an account from the list, the Customer 360° Information screen appears with the details of the respective account.	aid: 0586096368

If you want to search for a person or account using any additional search criteria, you can directly access the **Customer 360° View** screen by clicking the **Advanced Search** link in the **Search** field.

Characteristic Support for Deal Price List Assignment

Oracle Revenue Management and Billing enables you to assign a price list to a person or an account in a deal either from the user interface (i.e. through the **Price List Assignment** screen which is accessible from the **Deal Information** screen) or through the **C1-DealPricelistAssignment** or **C1-DealPricelistAssignmentREST** inbound web service. Now, you can also maintain characteristics for a price list assignment in a deal. The system enables you to define, edit, delete, and view the characteristics of a price list assignment in a deal from the user interface and through the **C1-DealPricelistAssignment** or **C1-DealPricelistAssignmentREST** inbound web service. To implement this feature, the following changes are made to the system:

- A new section named **Characteristics** is added in the **Assign Price List** section of the **Price List Assignment** screen (which is accessible from the **Deal Information** screen).
- The following tags are newly introduced in the **C1-DealPricelistAssignmentBS** business service schema:

Parent Tag	New Tags
<pricelistassignmentlist>	<characteristicsList> <characteristicType></characteristicType> <characteristicValue></characteristicValue> <effectiveDate></effectiveDate> </characteristicsList>

Note: Here, you can define those characteristic types where the characteristic entity is set to **Price List Assignment**.

Standard Pricing Comparison in Deal through Inbound Web Service

Oracle Revenue Management and Billing enables you to use the pricing from the standard price list of an entity during deal simulation to obtain an optimal pricing for a customer. Until now, if the **Standard Pricing Comparison** field was set to **Yes** in a deal type, the system used to calculate the following using the standard pricing while simulating the respective deals from the user interface:

- Deal level revenue, profit, profitability(%), revenue variation(%), and profit variation(%) using the standard pricing
- Product level revenue, profit, profitability(%), revenue variation(%), and profit variation(%) using the standard pricing
- Division level revenue, profit, profitability(%), revenue variation(%), and profit variation(%) using the standard pricing
- Revenue, profit, profitability(%), revenue variation(%), and profit variation(%) for each entity in the customer hierarchy using the standard pricing
- Revenue, profit, profitability(%), revenue variation(%), and profit variation(%) for each price item and pricing parameters combination using the standard pricing

You can then compare the financial summary of the deal using the original, proposed, standard, and recommended (if any) pricing. Now, the standard pricing comparison feature is also available when you simulate a deal through the **C1-Deal** or **C1-DealREST** inbound web service.

The following tags are newly introduced in the **C1-DealCreationUpdateIWS** business service schema:

Parent Tag	New Tags
<dealSimulationDetails>	<standardDealFinancialSummary> <revenue></revenue> <cost></cost> <profit></profit> <profitabilityPercentage></profitabilityPercentage> </standardDealFinancialSummary> <standardRevenueVariationPercentage></standardRevenueVariationPercentage> <standardProfitVariationPercentage></standardProfitVariationPercentage>
<divisionFinancialSummaryDetails>	<standardDealFinancialSummary> <revenue></revenue> <cost></cost> <profit></profit> <profitabilityPercentage></profitabilityPercentage>

Parent Tag	New Tags
	<pre> </standardDealFinancialSummary> <standardRevenueVariationPercentage></standardRevenue VariationPercentage> <standardProfitVariationPercentage></standardProfitVariati onPercentage> </pre>
<productFinancialSummaryDetails>	<pre> <standardDealFinancialSummary> <revenue></revenue> <cost></cost> <profit></profit> <profitabilityPercentage></profitabilityPercentage> </standardDealFinancialSummary> <standardRevenueVariationPercentage></standardRevenue VariationPercentage> <standardProfitVariationPercentage></standardProfitVariati onPercentage> </pre>
<hierarchyDetails>	<pre> <standardRevenue></standardRevenue> <standardCost></standardCost> <standardProfitabilityAmt></standardProfitabilityAmt> <standardProfitabilityPercentage></standardProfitabilityPer centage> <standardProfitVariationPercentage></standardProfitVariati onPercentage> <standardRevenueVariationPercentage></standardRevenue VariationPercentage> </pre>

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), 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.
	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 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 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						

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
			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 -30-Jun-2020	01-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	-	-

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.

New Features (Specific to Insurance)

This section describes the following new features added in this release which can be used in the health insurance domain:

- [Automatic Offset of Debit and Credit Bill Line Items](#)
- [To Do Notification on Binder Payment Cancellation](#)
- [Transfer Binder Payment from General Suspense Account to Individual Membership Account](#)
- [Reconciliation of Individual Health Insurance Subsidy Payments](#)
- [Reconciliation Discrepancy Report of Individual Health Insurance Subsidy Payments](#)
- [Deferred Processing Mode for Discrepancy Report Creation](#)
- [Invoice Request for Individual Health Insurance Billing](#)
- [Collection Class](#)
- [Delinquency Management](#)
- [Outbound Message Generation for Delinquency Management](#)
- [Delinquency Grace Period Business Rule](#)
- [Delinquency Event Attribute Business Rule](#)
- [Delinquency Termination Date Rule Business Rule](#)
- [Delinquency Miscellaneous Options Business Rule](#)
- [Hold Repricing](#)
- [Maintaining Address Status](#)
- [Prorate Membership Benefit and Corresponding Sponsor Charges](#)
- [Zero-Amount Membership Benefit Charges](#)
- [Additional Repricing Entity Detail Records](#)
- [Allocate Advance Deposit for Delinquent Customers](#)
- [Accounts Payable \(Refund Adjustments\) Extraction](#)
- [Updating Accounts Payable Request in ORMB](#)
- [Delinquency Process Type Preference](#)

Automatic Offset of Debit and Credit Bill Line Items

The fully insured individual business do come across various scenarios wherein the customers would need a facility to automatically offset the debit and credit bill line items of an account. For example,

- An individual enrolls for a health plan. But the individual fails to pay the binder payment before the membership start date. Such individual memberships are canceled once the appropriate inbound message is received to cancel the individual membership. On canceling the individual membership, an adhoc invoice is generated in the system and the premium billed for the individual membership is reversed. In this scenario, you may want to automatically offset the debit and credit bill line items of the individual member's account.

- An individual enrolls for a health plan having subsidy. The subsidy fully covers the premium, and therefore a zero balance bill is generated. If you internally offset a zero balance bill, the system matches the debit and credit line items of a zero balance bill on the bill completion. But, in this case, the match events are not created at the bill line item level. In such scenario, you may want to automatically offset the debit and credit bill line items of the individual member's account such that match events are created at the bill line item level.

Oracle Revenue Management and Billing facilitates you with the auto maintenance process for accounts wherein the system will automatically offset the debit and credit bill line items of the accounts. The auto maintenance process is driven using the **Offset Request** feature ensuring that the match events are created at the bill line item level. Note that, at present, the auto maintenance process for accounts is only tested and certified for the different lines of business in the health insurance domain.

While running the auto maintenance process for an account, the system creates an auto maintenance offset request for the account when either of the following criteria is met:

- Total amount of the open and unmatched items of the account is equal to zero. For example, an A1 account has two bills – B1 for Jan 2023 with the open balance 200 and B2 for Feb 2023 with the open balance -200 and thereby the A1 account balance is zero.
- Total amount of the open and unmatched items of the account in a specific coverage period is equal to zero. For example, an A2 account has three bills – B1 for Jan 2023 with the open balance 300, B2 for Feb 2023 with the open balance 300, and B3 for Feb 2023 with the open balance -300. Here, the overall account balance of A2 is not zero, but the account balance of A2 for Feb 2023 is zero.

Note: While running the auto maintenance process for an account, the system does not consider the payments applied on the On Account, Excess Credit, and Suspense contracts of the account.

The system creates an auto maintenance offset request for an account using an offset request type where the offset category is set to **Auto Maintenance**. A new field named **Offset Category** is available while defining an offset request type. It helps to differentiate between the traditional (i.e. manual) offset requests and automatic offset requests. If you want to create a manual offset request using an offset request type, you must set the offset category in the offset request type to **Regular**. However, if you want to create an automatic offset request using an offset request type, you must set the offset category in the offset request type to **Auto Maintenance**. By default, the offset category is set to **Regular**. Note that you cannot edit the offset category in an offset request type when an offset request is already created using the offset request type.

The system enables you to skip a set of accounts for which auto maintenance is not required. A new field named **SKIP_AUTOM_SW** is added in the **CI_ACCT** table. It indicates whether you want to skip auto maintenance for the account. At present, you can provide the skip auto maintenance information for an account through a healthcare inbound message and not from the user interface. The system then displays the skip auto maintenance information of an account in the **Main** section of the **Account Information** zone on the **Account** tab of the **Customer 360° Information** screen. Note that the **Skip Auto Maintenance** field appears in the **Main** section for the account only when it is set to **Y**.

To use the auto maintenance feature for accounts, you need to do the following:

- Create an offset request type where the offset category is set to **Auto Maintenance**

Note: While creating an offset request type for auto maintenance of accounts, you need to ensure that the **Bill Line Items Level Offset** option is selected and the **Transfer Adjustment** option is not selected.

- Set the **Avoid internal match event creation for zero dollar bill** option type of the **C1_FINTRANOP** feature configuration to **Y**

A new batch named **C1-AUTOM** is introduced in this release. It considers those accounts for auto maintenance where the **Skip Auto Maintenance** field is set to **N**. It then checks whether either of the following criteria is met for the account:

- Total amount of the open and unmatched items of the account is equal to zero
- Total amount of the open and unmatched items of the account in a specific coverage period is equal to zero

Here, the system considers the eligible accounts with zero and non-zero balance. If the eligible account has zero balance, the system considers the bills with zero and non-zero balance where the open and unmatched debit and credit line items can be offset. If the eligible account has non-zero balance, the system considers the bills with zero and non-zero balance where the open and unmatched debit and credit line items of the same coverage period can be offset.

If either of the above two criteria is met, the system creates and processes an auto maintenance offset request. The auto maintenance offset request automatically closes the open and unmatched bill line items of the account that amounts to zero balance. To close an open and unmatched bill line item of the account, the system creates an offset request adjustment against each such debit or credit financial transaction. If the match event is present on any financial transaction, the same match event is stamped on the offset request adjustment. However, if the match event does not exist on any financial transaction, the system creates a new match event and stamps it against the financial transaction and its corresponding offset request adjustment.

This batch is a multi-threaded batch. The multi-threading is based on account ID and chunks for multi-threading are created based on numerical distribution of account ID. This batch contains the following parameters:

Parameter Name	Mandatory (Yes or No)	Description
Division	Yes (Conditional)	Used when you want to run the auto maintenance process for the accounts which belong to a particular division. Note: This parameter is required when you specify a customer class while executing this batch.
Customer Class	No	Used when you want to run the auto maintenance process for the accounts which belong to a particular customer class.

Parameter Name	Mandatory (Yes or No)	Description
Offset Request Type	Yes	Used to indicate the offset request type using which you want to create auto maintenance offset requests for the accounts. Note: Here, you must specify an offset request type where the offset category is set to Auto Maintenance .
Freeze Number of Days	Yes	Used to specify the number of days. The system considers those accounts for auto maintenance whose financial transactions are frozen in the last specified number of days. Note: The parameter value must be greater than or equal to zero.
Skip Auto Maintenance for Accounts with Open Refund or Write Off Request (Y/N)	Yes	Used to indicate whether you want to skip auto maintenance for those accounts for which the refund or write off request is still in the non-final status. The valid values are: <ul style="list-style-type: none"> • Y • N Note: By default, the parameter value is set to Y .
Skip Auto Maintenance for Accounts with Open Payment, Payment Transfer Request (Y/N)	Yes	Used to indicate whether you want to skip auto maintenance for those accounts for which the payment creation or payment transfer request is still in the non-final status. The valid values are: <ul style="list-style-type: none"> • Y • N Note: By default, the parameter value is set to Y .
Skip Auto Maintenance for Accounts with Open Offset Request (Y/N)	Yes	Used to indicate whether you want to skip auto maintenance for those accounts for which the manual offset request is still in the non-final status. The valid values are: <ul style="list-style-type: none"> • Y • N Note: By default, the parameter value is set to Y .
Account ID	No	Used when you want to run the auto maintenance process for a particular account.

Parameter Name	Mandatory (Yes or No)	Description
Fetch Only Open Overdue Bills	Yes	<p>Used to indicate whether you want to consider the overdue bills or all the open bills of the account for auto maintenance. The valid values are:</p> <ul style="list-style-type: none"> • Y – Used when you want to consider only the overdue bills of the account for auto maintenance. • N – Used when you want to consider all the open bills of the account for auto maintenance. <p>Note: By default, the parameter value is set to Y.</p>
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

On successful completion of this batch, the system creates and processes an auto maintenance offset request for an account that meets the criteria. You can view the details of an auto maintenance offset request through the **Offset Request** screen.

While viewing the details of an auto maintenance offset request, the system displays the **Auto Maintenance Financial Transactions** zone instead of the **Selected Bills** and **Offset Request Adjustments** zones in the **Main** tab of the **Offset Request** screen. The system also displays the account information in the **Main** section of the **Offset Request** zone. The **Auto Maintenance Financial Transactions** zone lists the financial transactions of the account for which the corresponding offset request adjustments are created during the auto maintenance process.

To design the auto maintenance process for accounts, the following algorithms are enhanced:

- **C1-OFFSUBENT** – Here, some of the existing validations are skipped for an auto maintenance offset request and few new validations are introduced for an auto maintenance offset request.
- **C1-OFST-DIST** – Here, additional logic is added to handle the offset category and to create auto maintenance offset requests for accounts that meet the criteria. Note that the **Match Type**, **Match Type Entity**, and **Credit Bill Distribution Algorithm** parameters are not applicable while creating offset request adjustments for an auto maintenance offset request.

The system enables you to cancel an auto maintenance offset request, if required, through the **Unapply** button. While canceling an offset request, you need to specify a reason why you want to cancel an offset request. On canceling an auto maintenance offset request, the system cancels the offset request adjustments and opens the corresponding bill line items which were offset during auto maintenance of the account.

To Do Notification on Binder Payment Cancellation

Oracle Revenue Management and Billing enables you to generate a To Do notification when a binder payment for an individual membership is canceled. To generate the To Do notification when a binder payment is canceled, you need to do the following setup:

- Add the following value in the **MEMB_PERS_STAT_RSN_FLG** lookup field:
 - Binder Payment Canceled
- Add the following status reason for the **Active** status of the **C1-IndMembership** business object:
 - Membership is active but binder payment is canceled
- Add the following the status reason mapping in the preference which is defined using the **Membership Status Reason Mapping (C1-MemStatusReasonMapping)** business object:

Member Person Status	Member Person Status Reason	Membership Status Reason
Active	Binder Payment Canceled	Membership is active but binder payment is canceled

- Define the following attributes in the preference which is defined using the **C1-FieldMapping (C1-FieldMapping)** business object and where the preference category is set to **Binder Payment**:

Attribute	Value	Entity Type
Membership Status Reason when Binder Payment Cancelled	Membership Status Reason Code (of the Membership is active but binder payment is canceled status)	Membership
To Do Type for Binder Payment Cancelled	C1-BPCNC	Membership

A new To Do type named **C1-BPCNC** and algorithm type named **C1-BNDPYCNCL** are introduced in this release. To generate the To Do notification during binder payment cancelation, you need to attach an algorithm created using the **C1-BNDPYCNCL** algorithm type to the **Payment Cancellation** system event of the required customer class. If the **C1-BNDPYCNCL** algorithm is attached on the customer class, the system checks whether the cancel reason specified while canceling a payment is listed as a value of the **Binder Payment Cancel Reason List** parameter in the algorithm. If so, the system then validates whether the payment is a binder payment using the **Binder Payment Identification Characteristics Type** attribute of the binder payment preference. If the payment is a binder payment, the system then determines the individual membership whose binder payment is canceled.

Points to Note:

If a payment contains a membership identifier value characteristic that is specified in the **Binder Payment Identification Characteristics Type** attribute of the binder payment preference, the system considers the payment as a binder payment.

The system derives the individual membership using the membership identifier value characteristic on the binder payment and the **Binder Payment Identification ID Type** attribute of the binder payment preference.

Once the individual membership is derived, the system fetches the binder payment applicability flag for the individual membership. If the binder payment applicability flag of the individual membership is set to **No**, the system will not generate the To Do notification when the binder payment is canceled.

However, if the binder payment applicability flag of the individual membership is set to **Yes**, the system will derive the binder payment threshold amount for the respective membership. If the binder payment threshold amount is equal to zero, the system will not generate the To Do notification when the binder payment is canceled.

If the binder payment threshold amount is greater than zero, the system will calculate the grace date (Membership Start Date + (Binder Payment Grace Days -1)) and the total binder payment amount until the grace date. If the **Consider Binder Liability Amount** attribute in the binder payment preference is set to **Y**, the system checks whether the total binder payment amount until the grace date is less than the threshold amount. If the total binder payment amount until the grace date is less than the threshold amount, the system does the following:

- Updates the member person status reason to **Binder Payment Canceled** and the membership status reason to **Membership is active but binder payment is canceled**.
- A To Do using the **C1-BPCNC** To Do type is created.

However, if the total binder payment amount until the grace date is greater than or equal to the threshold amount, the system neither changes the member person and membership status reasons nor creates a To Do notification when the binder payment is canceled.

If the **Consider Binder Liability Amount** attribute in the binder payment preference is set to **N**, the system checks whether the total binder payment amount until the grace date is equal to zero. If the total binder payment amount until the grace date is equal to zero, the system does the following:

- Updates the member person status reason to **Binder Payment Canceled** and the membership status reason to **Membership is active but binder payment is canceled**.
- A To Do using the **C1-BPCNC** To Do type is created.

However, if the total binder payment amount until the grace date is greater than zero, the system neither changes the member person and membership status reasons nor creates a To Do notification when the binder payment is canceled.

This algorithm contains the following parameters:

- **Membership Business Object** – Used to specify the individual membership business object. The system will then consider only those individual memberships which are created using the specified business object. Here, you must set the value to **C1-IndMembership**.
- **Algorithm for TO DO Creation** – Used to specify the algorithm using which you want to create a To Do notification during binder payment cancellation. Here, you must set the value to an algorithm which is created using the **C1-BNDTODO** algorithm type.
- **Binder Payment Cancel Reason List** – Used to specify a comma-separated list of payment cancel reasons that the system should consider while To Do generation on payment cancellation. You can specify maximum five comma-separated values for this parameter. Here, you must specify a payment cancel reason which is already defined in the system.

All the above parameters are mandatory.

Note that the system will update the membership status to **Pending Effectuation** only when the inbound message is received to change the membership status to **Pending Effectuation**. Until then, the member person and membership statuses will not change even if the respective binder payment is canceled.

Transfer Binder Payment from General Suspense Account to Individual Membership Account

Usually, in a fully insured individual business, the binder payments received before the individual membership enrollment are parked in the general suspense account. Oracle Revenue Management and Billing enables you to transfer the binder payments of an individual membership from the general suspense account to the individual member's account after the individual membership enrollment information is received through an inbound message.

To transfer the binder payment from general suspense account to individual membership account, you need to set the value for the following option type in the **C1_CMO** feature configuration:

- **Suspense Contract Type** – Used to indicate the suspense contract type whose suspense contracts' payments you want to transfer to the individual membership account.

A new batch control named **C1-PYTRS** is introduced in this release. It fetches the suspense contracts of the contract type which is specified in the **Suspense Contract Type** option type of the **C1_CMO** feature configuration. Once all suspense contracts are derived, the batch fetches the unmatched frozen payments which are made against the suspense contracts. The batch then checks whether the characteristic type specified in the **Payment Identifier Characteristic Type** parameter is defined on the payment. If the specified characteristic type is defined on the payment, the system considers the payment as the binder payment. If the specified characteristic type is not defined on the payment, the system does not consider the payment as the binder payment.

The batch then derives the individual membership for which the binder payment is made. While deriving the individual membership, the batch uses the membership identifier value characteristic defined on the binder payment and the value specified in the **Membership Identifier Type** parameter. You can specify a comma-separated list of membership identifier types in the **Membership Identifier Type** parameter. The batch then checks whether an individual membership exists with either of the membership identifier type and value combination in the system. Once the individual membership is derived, the batch fetches the individual member's account using the account identifier type and value characteristics which are defined on the individual membership.

Points to Note:

While deriving the account identifier type and value characteristics, the system considers the characteristic types which are specified in the **Account Identifier Type Char Type** and **Account Identifier Value Char Type** option types of the **C1-ASOBLNG** feature configuration.

If multiple individual memberships are derived with the membership identifier type and value combination, the system will not consider the binder payment for transfer and will accordingly add an appropriate log entry to such binder payments.

Once the individual member's account is derived, the batch transfers the binder payment from the suspense contract to either the unpaid bills or on account contract of the account depending on the following:

- If the match type is specified in the **Bill Match Type** parameter, the batch will transfer the binder payment from the suspense contract against the unpaid bills of the account. While transferring the binder payment against the unpaid bills of the account, the system will do either of the following:
 - If the **C1-MD-ACCT** manual distribution algorithm is attached to the match type, the system will distribute the binder payment against the unpaid bills of the account in the ascending order of the due date. If there are multiple bills with the same due date, the system will consider the unpaid bill in the ascending order of the unpaid amount. However, if there are multiple bills with the same due date and amount, the system will randomly consider the unpaid bill for payment. If there is any excess credit, the system will apply the excess credit amount on the on account contract of the individual member's account. Note that the on account contract is derived using the match type specified in the **On Account Match Type** parameter.
 - If a manual distribution algorithm is not attached to the match type, the system will distribute the binder payment against the unpaid bills of the account in the ascending order of the due date. If there is any excess credit, the system will apply the excess credit amount on the on account contract of the individual member's account. Note that the on account contract is derived using the match type specified in the **On Account Match Type** parameter.
- If the match type is not specified in the **Bill Match Type** parameter, the batch will transfer the binder payment from the suspense contract to the on account contract of the individual member's account. Note that the on account contract is derived using the match type specified in the **On Account Match Type** parameter.

While transferring the binder payment from the suspense contract to either the unpaid bills or on account contract, the system cancels the old payment on the suspense contract and creates a new payment against the unpaid bills or on account contract of the individual member's account. While canceling the old payment, the system uses the cancel reason specified in the **Payment Cancel Reason for Transfer** parameter. While creating a new binder payment, the system copies all characteristics from the old payment before changing its status to **Frozen**.

This batch is a multi-threaded batch. The multi-threading is based on account ID and chunks for multi-threading are created based on numerical distribution of account ID. This batch contains the following parameters:

Parameter Name	Mandatory (Yes or No)	Description
Payment Identifier Characteristic Type	Yes	Used to specify the characteristic type which stores the membership identifier value. The system then considers a payment as a binder payment if the specified characteristic type is defined on the payment. Note: You must specify the characteristic type where the characteristic entity is set to Payment .
Membership Identifier Type	Yes	Used to indicate the membership identifier type using which you want to derive the individual membership. If you specify multiple membership identifier types separated by comma, the system will derive the individual membership using either of the membership identifier type and value combinations. Points to Note: You must specify a membership identifier type which is already defined in the system. You can specify maximum twenty comma-separated values for this parameter.
Bill Match Type	No	Used to indicate the match type using which you want to transfer the binder payment from the suspense contract to the unpaid bills of the individual member's account. Note: You must specify a match type where the entity type is set to Account .
On Account Match Type	Yes	Used to indicate the match type using which you want to transfer the binder payment from the suspense contract to the on account contract of the individual member's account.

Parameter Name	Mandatory (Yes or No)	Description
Payment Cancel Reason for Transfer	Yes	Used to indicate the payment cancel reason that you want to use while canceling the binder payment on the suspense contract. Note: You must specify a payment cancel reason which is already defined in the system.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

Once the binder payment is transferred from the suspense contract to either the unpaid bills or on account contract of the individual membership's account, the payor account details on the tender is updated to the individual membership's account. Let us understand this with the help of an example.

The following table lists the details of a binder payment which is made against the general suspense account:

Payment Event ID	Account	Payment Amount	Payment Char Type	Payment Char Value	Payment Date	Account Type
PAY_ID1	SUS ACT 1	300	Payment Reference ID	M001	01-Nov-2023	General Suspense Account

Now, when you execute the **C1-PYTRS** batch after the individual membership inbound is received in ORMB, the system derives the individual membership for which binder payment is made and transfers the binder payment from the general suspense account to the individual membership's account. The following table illustrates two payments – one on the general suspense account and another on the individual membership account.

Payment Event ID	Payment ID	Account	Payment Amount	Payment Char Type	Payment Char Value	Payment Date	Payment Match Entity	Status
PAY_ID1	P1	SUS ACT 1	300	Payment Reference ID	M001	01-Nov-2023	General Suspense Account	CANCEL LED
PAY_ID1	P2	ACT 1	300	Payment Reference ID	M001	01-Nov-2023	Member Account	FROZEN

Once the payment is created on the individual membership's account, the system updates the payor account on the tender from SUS ACT 1 to ACT 1.

Reconciliation of Individual Health Insurance Subsidy Payments

Several states have established health insurance exchanges to offer health and dental insurance coverage options to their respective residents. Individuals within these states can compare and purchase coverage from a selection of health and dental plans, which are referred to as Qualified Health Plan (QHP).

Individuals who enroll for these QHPs may be eligible for Federal Tax Credits and Cost-Sharing Reductions (CSRs), which are subsidies based on the subscriber's income, to help pay for their health insurance and lower their out-of-pocket costs. The Centers for Medicare & Medicaid Services (CMS) makes monthly payments to insurers for the individuals who receive these credits.

CMS has established and instituted the State-Based Exchange (SBE) or Federally Facilitated Marketplace (FFM) reconciliation process to ensure that insurers enrollment information aligns with Marketplace enrollment information. The dispute process provides a mechanism for insurers to correct an FFM enrollment record or related payment information that the monthly reconciliation process cannot resolve.

Oracle Revenue Management and Billing (ORMB) may receive either the exact or different payment amount from the exchange for the sponsored charges due to various reasons. ORMB now enables you to reconcile the payments from the exchange against the bill line items for the fully insured individual business, thereby ensuring that the discrepancy is identified and resolved on time to quickly recover the money from CMS via the dispute process.

The reconciliation process in ORMB is enhanced to support both the fully insured group business and fully insured individual business. For the fully insured group business, the system stamped the policy ID and plan ID in the **C1_FT_EXT** table while freezing a financial transaction on the bill completion. These details were later used to reconcile the pay instructions against the bill line items. In this release, the **C1-STMPFTINF** algorithm type (which is attached on the **FT Freeze** system event of a customer class) is enhanced to support the fully insured individual business. It stamps the health plan code and price item in the **C1_FT_EXT** table while freezing a financial transaction on the bill completion for the fully insured individual business. These details are later used to reconcile the pay instructions received from the exchange against the bill line items of the exchange account.

A new field named **Reconciliation Category** is available while defining a reconciliation type. It helps to differentiate between the fully insured group and fully insured individual reconciliation objects. If you want to upload and process a pay instruction file for a fully insured group business using a reconciliation type, you must set the reconciliation category of the reconciliation type to **Group**. However, if you want to upload and process a pay instruction file for a fully insured individual business using a reconciliation type, you must set the reconciliation category of the reconciliation type to **Individual**.

A new algorithm type named **C1-PAYINSEXC** is introduced in this release. You need to attach an algorithm created using the **C1-PAYINSEXC** algorithm type to the **Upload Pay Instructions** system event of a reconciliation type where the reconciliation category is set to **Individual**. This algorithm parses a pay instruction file which is received from the exchange. It reads each record in the pay instruction file and creates one or more pay instructions in a reconciliation object depending on the number of exchange payment types received in the record. For example, if a record contains information about two exchange payment types (i.e. EP1 and EP2), the system will create two pay instructions for the record – one for EP1 and another for EP2. The exchange payment types (i.e. health insurance coverage) maintained in the CMS system might be different from the price items maintained in ORMB. Therefore, ORMB enables you to map an exchange payment type with a price item using the **C1-PayTypePrcltemMap** extendable lookup. Note that no values are shipped for this extendable lookup from the product. You need to create a value for the **C1-PayTypePrcltemMap** extendable lookup wherein each payment type received from the exchange is mapped to a price item. You can map a price item to one or more payment types based on the requirements.

While creating an algorithm using the **C1-PAYINSEXC** algorithm type, you need to specify the following parameters:

- **Pay Instruction Business Object** – Used to indicate the business object using which you want to create a pay instruction.
- **Date Format** – Used to indicate the format in which you want to store the coverage period start and end dates.
- **Payment Type – Price Item mapping** – Used to indicate the **C1-PayTypePrcltemMap** extendable lookup value using which you want the system to derive the price item for each payment type received from the exchange.

The system enables you to upload a pay instruction file received from the exchange in the CSV file format. You can upload pay instruction files in the CSV format from the specified location on the server using the **Pay Instruction CSV File Upload (C1-RECUP)** batch. While receiving a pay instruction file from the exchange, you need to ensure that each record in the pay instruction file contains the following information:

- Payor (i.e. exchange person) or its account from where the payment is received
- Subscriber (i.e. individual membership) for whom the payment is received
- Health plan and its coverage period for which the payment is received
- At least one exchange payment type (i.e. health insurance coverage) and its subsidy amount which is sponsored by the exchange

Points to Note:

The system assumes that the exchange person will have only one account. Therefore, if an exchange person has multiple accounts, the system will not allow you to upload and process a pay instruction file received from the respective exchange. Hence, if the exchange person has a single account in the system, you can either provide the payor or its account details in the record. However, if the exchange person has multiple accounts in the system, you must provide the payor's account details in the record.

You can create an exchange person and its account in the system through the health care inbound message. While creating an account for an exchange person, ensure that you set the **Eligible for Member Reconciliation (C1-RCELG)** characteristic of the account to **Y**.

You can specify either the health plan code or Health Insurance Oversight System (HIOS) ID in the record. If you are planning to use the Health Insurance Oversight System (HIOS) ID for a health plan, you need to attach an algorithm created using the **C1-STMPHLPN** algorithm type to the **Enter** system event (with the lowest sequence number) of the **Pending** status in the **C1-MemberReconciliation** business object. This algorithm will then derive the code of the health plan, to which the individual is enrolled for the coverage period, using the HIOS ID. Note that, while creating a pay instruction, the system will derive the health plan code when the HIOS ID is present in the record irrespective of whether the health plan code is present in the record or not. However, if the HIOS ID is not present in the record, the system will use the health plan code that is present in the record.

If a subscriber has enrolled for multiple insurance coverages of a health plan for the same coverage period and if the subscriber is eligible for subsidy for one or more insurance coverages, you will receive the subsidy amount for the respective insurance coverages from the CMS system. The subsidy information is received in the form of exchange payment type and amount. At a time, you can specify maximum 10 exchange payment types and their amounts in each record of a pay instruction file.

The **C1-RECUP** batch is enhanced to support the fully insured individual business. On executing the **C1-RECUP** batch, the system reads the pay instruction file from the specified location and validates it. Once a file is successfully validated, the reconciliation object is created for the file in the **Draft** status. The reconciliation object is immediately transitioned to the **Send Notification** status and the algorithms attached to the **Send Notification** status are executed. Once the To Do is created, the status of the reconciliation object is changed to **Pending**. One or more pay instructions are created for a record of a pay instruction file and its status is set to either **Pending** or **Error** depending on whether it is successfully validated or not. While creating a pay instruction, the system derives the health plan code (if not available), individual membership and the exchange account. For more information about the **C1-RECUP** batch, refer to *Oracle Revenue Management and Billing Batch Guide*.

You need to then specify the payment ID against which you want to reconcile the bill line items for which you have received the pay instructions. Once you specify the payment information and submit the pay instructions for reconciliation, the payment amount is distributed against the reconciliation contract of the exchange accounts for which you have received the pay instruction. The status of the reconciliation object is changed to **Pending Reconciliation**. On reconciling the pay instructions, the system finds the open and unmatched bill line item of the exchange account against which the pay instruction must be reconciled using the health plan, subscriber, price item, and coverage period combination. The system creates a reconciliation adjustment for a pay instruction which is successfully reconciled. While reconciling the pay instructions, the system sets the pay instruction matching level characteristic type of the reconciliation object to **MEMBER**.

You can manually reconcile the pay instructions of a reconciliation object, or you can execute the **Reconciliation Periodic Monitor (C1-RCNM)** or **Pay Instruction Processing Batch (C1-RCPM)** batch at regular intervals to reconcile the pay instructions.

Note: While executing the **C1-RCPM** batch, you need to specify a member reconciliation preference whose attributes you want to use while reconciling all pay instructions of a file at the member level.

For more information about the above batches, refer to *Oracle Revenue Management and Billing Batch Guide*.

If you attach an algorithm created using the **C1-VALMRPYCN** algorithm type to the **Payment Cancellation** system event of the required customer class, the system automatically cancels the payment and reconciliation object associated with the payment when you cancel its corresponding payment tender. The status of the reconciliation object is changed to **Pending Cancelation** and the status of all pay instructions in the reconciliation object is changed to **Canceled**.

If required, you can also manually cancel a reconciliation object. However, you can cancel a reconciliation object only when it is in the **Open** or **Completed** status. On canceling a reconciliation object, the status of the reconciliation object is changed to **Pending Cancelation**. Before canceling a pay instruction, the reconciliation adjustment (if any) corresponding to the pay instruction is also canceled.

You need to configure the **Reconciliation Cancellation Periodic Monitor (C1-RCNMD)** batch to execute at the regular intervals. This batch monitors whether there are any reconciliation objects in the **Pending Cancelation** status. If there is a reconciliation object in the **Pending Cancelation** status, the status of the reconciliation object and the status of all pay instructions in the reconciliation object is changed to **Canceled**. For more information about the **C1-RCNMD** batch, refer to *Oracle Revenue Management and Billing Batch Guide*.

If all pay instructions in a reconciliation object are successfully reconciled, the system changes the status of the reconciliation object to **Completed**. However, if one or more pay instruction in a reconciliation object is not successfully reconciled, the system changes the status of the reconciliation object to **Open**. If you want the system to automatically reconcile the open pay instructions of the account when you bill the account each time, you need to attach an algorithm created using the **C1-RCLOPNRCN** algorithm type to the **Post Bill Completion** system event of the required customer class. The system will then trigger the reconciliation process to reconcile the open pay instruction against the open and unmatched bill line item of the exchange account and will accordingly update the status of the pay instruction and reconciliation object.

If you attach an algorithm created using the **C1-VLPYINACN** algorithm type to the **Adjustment Cancellation** system event of the adjustment type (using which the reconciliation adjustments are created), the system ensures that no one can cancel a reconciliation adjustment until it is linked to a non-canceled pay instruction.

Reconciliation Discrepancy Report of Individual Health Insurance Subsidy Payments

Until now, you were able to generate the reconciliation discrepancy report at the file or account level for the fully insured group business. Now, in addition, you can generate the reconciliation discrepancy report at the file or account level for the individual health insurance subsidy payments. If you generate the discrepancy report at the file level, the system lists the following:

- All pay instructions in the file which were partially reconciled due to rate variance
- All pay instructions in the file which could not be reconciled as the amount was paid, but not yet billed

However, if you generate the discrepancy report at the account level, the system lists the following:

- All bill line items which were partially reconciled against pay instructions due to rate variance
- All bill line items which could not be reconciled as the amount was billed, but not yet paid
- All pay instructions for the account which could not be reconciled as the amount was paid, but not yet billed

While generating a discrepancy report, you need to specify the discrepancy report type using which you want to create the discrepancy report. The system will create the discrepancy report in either **Pending** or **Deferred Processing** status. Once you review the discrepancy line items and finalize the discrepancy report, its status is changed to **Completed**.

At a time, you can only have one discrepancy report for a file or an account in the **Pending** or **Deferred Processing** status. Until you finalize a discrepancy report, you cannot create another discrepancy report for the respective file or account. Also, note that you can create a discrepancy report for a file only when its corresponding reconciliation object is in the **Open** status.

When you generate a discrepancy report for a pay instruction file which is received from the exchange or when you generate a discrepancy report for an exchange account, the system displays the following additional information for each discrepancy line item along with other details:

- Payor Identifier Type
- Payor Identifier
- Health Plan
- HIOS ID
- Payment Type

Note: Policy number and plan number are displayed instead of the above information when you generate discrepancy report for the fully insured group business.

Deferred Processing Mode for Discrepancy Report Creation

Until now, the reconciliation discrepancy report at the file or account level was only created in the online mode (i.e. immediately). Now, the system provides you with the ability to configure deferred processing mechanism for the reconciliation discrepancy report creation. A new status named **Deferred Processing** is introduced in the lifecycle of the **C1-DiscrepancyReport** business object. On generating a discrepancy report at the file or account level, the discrepancy report is created in the **Draft** status. A new algorithm named **C1-DISCDEF** is attached to the **Enter** system event of the **Draft** status. This algorithm checks whether the number of pay instructions in the non-final status exceeds the deferred count specified in the algorithm. If the number of pay instructions in the non-final status does not exceed the deferred count, the status of the discrepancy report is changed to **Pending**. The system then adds the discrepancy line items in the discrepancy report by comparing each non-final pay instruction against the open and unmatched bill line item of the exchange account.

However, if the number of pay instructions in the non-final status exceeds the deferred count, the status of the discrepancy report is changed to **Deferred Processing**. A new batch named **C1-DRMT** is introduced in this release. This batch is used to monitor or check whether there are any discrepancy reports in the **Deferred Processing** status. If there is a discrepancy report in the **Deferred Processing** status, the batch changes the status of the discrepancy report to **Pending**. Then, the system adds the discrepancy line items in the discrepancy report by comparing each non-final pay instruction against the open and unmatched bill line item of the exchange account.

This batch is a multi-threaded batch. The multi-threading is based on discrepancy request ID and chunks for multi-threading are created based on numerical distribution of discrepancy request ID. This batch contains the following parameters:

Parameter Name	Mandatory (Yes or No)	Description
Maintenance Object	Yes	Used to indicate that you want to monitor discrepancy reports which are created using the business objects of a particular maintenance object. Note: By default, the parameter value is set to C1-DISCR-RPT .

Parameter Name	Mandatory (Yes or No)	Description
Restrict by Batch Code	No	Used when you want to monitor discrepancy reports whose current status is linked to the Discrepancy Report Monitor (C1-DRMT) batch. The valid value is true .
Restrict by Discrepancy Report Type	No	Used when you want to monitor the discrepancy reports which are created using a particular discrepancy report type.
Restrict by Business Object	No	Used when you want to monitor discrepancy reports which are created using a particular business object.
Restrict by Status Code	No	Used when you want to monitor discrepancy reports which are in a particular status. Note: This parameter is useful when this batch is invoked from more than one status in the lifecycle of the business object.
Override maximum errors	No	Used to override the maximum number of errors after which the batch must be terminated.
Threadpool	No	Used to specify the thread pool on which you want to execute the batch.

Invoice Request for Individual Health Insurance Billing

Until now, the system enabled you to generate an adhoc regular or trial bill for an account through an invoice request for the fully insured group business. Now, this feature is extended to support the adhoc billing requirements of the fully insured individual business.

The system enables you to manually or automatically create an invoice request for the fully insured individual business. It enables you to create an invoice request for an account to which an individual membership is billed. Before creating an invoice request for an account of an individual membership, the system validates whether the individual membership is eligible for the invoice request creation. You can create an invoice request only for an account of those individual memberships where the **Eligible For Invoice Request Creation (INVRACTE)** characteristic is set to **Y**. Whenever an invoice request is created for the account of an individual membership, the system adds the appropriate log entries for the individual membership in the **Log** tab of the **Membership** screen.

Note: The system validates whether the individual membership is eligible for invoice request creation using the characteristic type which is specified in the **Invoice Request Eligibility Char Type** option type of the **C1-INVREQ** feature configuration. By default, this option type is set to **INVRACTE**. You can change the characteristic type, if required.

The system enables you to create the following types of invoice request for the fully insured individual business:

- **Manual Regular Bill Generation Invoice Request** – It is used to generate an adhoc regular bill for an account. You can create a manual regular bill generation invoice request for an account from the user interface.
- **Manual Trial Bill Generation Invoice Request** - It is used to generate an adhoc trial bill for an account. You can create a manual trial bill generation invoice request for an account from the user interface.
- **Automatic Regular Bill Generation Invoice Request** – It is used to generate an adhoc regular bill for an account. You can configure the system such that an automatic regular bill generation invoice request is automatically created for an account when an individual membership is enrolled, terminated, reinstated, or canceled through a health care inbound message.

Points to Note:

The automatic invoice request creation feature is only designed to work for the fully insured group and individual businesses and not for the financial services or self-funded health insurance business.

At present, the system is not designed to create an automatic trial bill generation invoice request.

While creating an invoice request, you need to specify an invoice request type using which you want to create the invoice request. The system enables you to create the invoice request types with the following combinations:

Invoice Mode	Bill Generation Type	Approval Process
Manual	Regular	On
Manual	Regular	Off
Manual	Trial	Off
Automatic	Regular	Off

It is the invoice request type which helps the system to determine:

- The business object using which the invoice request should be created in the system

Note: If the **Bill Generation Type** field is set to **Regular**, you must select the **Invoice Request - Regular Bill Generation** business object from the **Invoice Request Business Object** list. However, if the **Bill Generation Type** field is set to **Trial**, you must select the **Invoice Request - Trial Bill Generation** business object from the **Invoice Request Business Object** list.

- Whether a manual or automatic invoice request should be created in the system
- Whether a regular or trial bill generation invoice request should be created in the system
- Whether a manual regular bill generation invoice request for an account should be processed in the online or deferred mode

Points to Note:

The system will process a manual regular bill generation invoice request for an account in the deferred mode when the number of billable charges of the account exceeds the defer processing billable charge count.

You can specify the defer processing billable charge count only when the invoice mode is set to **Manual** and bill generation type is set to **Regular**.

- Whether the approval is required for the invoice request

Note: You can set the **Approval Required** flag only when the invoice mode is set to **Manual** and bill generation type is set to **Regular**.

- The wait days using which the processing date is calculated for an invoice request

Note: If the invoice request is processed in the online mode, the system will process the invoice request when the processing date is earlier than or equal to the system date. However, if the invoice request is processed in the deferred mode, the system will process the invoice request when the processing date is earlier than or equal to the batch business date.

- The date calculation algorithm which is used to calculate the cutoff date, bill date, processing date, and accounting date for an automatic regular bill generation invoice request

Points to Note:

The **C1-INVRDTCIM** algorithm is shipped in this release for the fully insured individual business. It is used to calculate the cutoff date, bill date, processing date, and accounting date for an automatic regular bill generation invoice request. If required, you can configure the algorithm such that the processing date does not fall on a holiday. In that case, you need to specify the work calendar which helps to determine the working day. You can specify the work calendar in the **C1-INVRDTCIM** algorithm. Alternatively, if the work calendar is not specified in the **C1-INVRDTCIM** algorithm, the system will use the work calendar of the account's division to ensure that the processing date falls on a working day.

In addition, you can configure threshold days for the automatic invoice request creation. The system then determines whether the number of days between the bill cycle date and processing date is greater than the threshold days. If so, the automatic invoice request is created for the account. However, if the number of days between the bill cycle date and processing date is less than or equal to the threshold days, the automatic invoice request is created for the account such that cutoff date is calculated using the next window of the bill cycle.

While creating a manual invoice request, you need to specify the following – invoice request type, account, cutoff date, accounting date, and bill date. Note that before manually or automatically creating an invoice request for an account, you need to ensure that the following algorithms are attached to the respective system event of the corresponding customer class:

System Event	Algorithm
Invoice Request Account Selection	C1-INVACTEL
Invoice Request Status Update	C1-INVRSTUP

Manual Regular Bill Generation Invoice Request

Once you manually create a regular bill generation invoice request for an account, the status of the invoice request is set to **Draft**. You can then submit the invoice request. You can optionally configure the approval process for the manual regular bill generation invoice request. If the **Approval Required** flag is set to **Yes** in the invoice request type, the status of the manual regular bill generation invoice request is changed to **Approval In Progress** when you submit the invoice request. The approver can then approve or reject the manual regular bill generation invoice request based on the observations.

On submitting or approving the invoice request, the system checks whether the number of the billable charges of the account exceeds the defer processing billable charge count. If the number of the billable charges of the account exceeds the defer processing billable charge count, the status of the invoice request is changed to **Defer Processing**. However, if the number of the billable charges of the account does not exceed the defer processing billable charge count, the system checks whether the processing date of the invoice request is earlier than or equal to the system date.

If the processing date is earlier than or equal to the system date, the system checks whether the pending bill already exists for the account. If the pending bill does not exist for the account, the system checks whether the cutoff date is later than the bill after date. If cutoff date is later than the bill after date, the system checks whether the accounting calendar is open for the accounting date. If the accounting calendar is open for the accounting date, the system creates the adhoc regular bill for the account in the online mode. Finally, the status of the invoice request is changed to **Processed**. But, if the pending bill exists for the account, or if the cutoff date is earlier than or equal to the bill after date, or if the accounting calendar is not open for the accounting date, or if an error occurs while creating adhoc regular bill or its bill segments, the status of the invoice request is changed to **Error**. The system enables you to move such erroneous invoice request back to the **Draft** status.

However, if the processing date is later than the system date, the status of the invoice request is changed to **Defer Processing**. On the **C1-INVRQ** batch execution, the system checks whether there are any invoice requests in the **Defer Processing** status. If there is an invoice request in the **Defer Processing** status, the system checks whether the processing date of the invoice request is earlier than or equal to the system date. If the processing date is earlier than or equal to the system date, the status of the invoice request is changed to **Processing** and the algorithm attached to the **Processing** status is invoked. However, if the processing date is later than the system date, the status of the invoice request remains as **Defer Processing**.

Note: The system enables you to cancel an invoice request which is in the **Defer Processing** status.

Manual Trial Bill Generation Invoice Request

The manual regular bill generation invoice request and manual trial bill generation invoice request are created and processed identically with the following exceptions:

- On submitting the manual trial bill generation invoice request for an account, the system does not check whether the number of the billable charges of the account exceed the defer processing billable charge count. The manual trial bill generation invoice request is always processed in the deferred mode.
- You need to specify the following parameters while executing the **BILLOPEN**, **BSGENREG**, and **POSTPROC** batches:
 - **Bill Generation Type** (set to **Trial**)
 - **Description for Trial Billing Batch Run**
 - **Off Cycle Switch** (set to **Y**)
- The **BSGENREG** batch will not freeze and complete a trail bill for the account.
- At present, a To Do is not generated for a trial bill review, and therefore the system does not check whether a To Do is generated for a trial bill review.

Automatic Regular Bill Generation Invoice Request

To enable the automatic invoice request creation feature for the fully insured individual business, you need to attach the following algorithms:

If you want to create an automatic invoice request for the account while...	Then attach the algorithm....			
	Business Object	Status	System Event	Algorithm
Activating an individual membership	C1-IndMembership	Active	Enter	C1-INVRQIMEN
Terminating an individual membership		Terminated	Enter	C1-INVRQIMTM
Reinstating a terminated individual membership		Active	Enter	C1-INVRQIMEN
Canceling an active individual membership		Canceled	Enter	C1-INVRQIMCN

You can create a single or different invoice request type for each individual membership event (such as, activation, termination, reinstatement, or cancelation) depending on the wait days you want to configure for the respective event. While attaching the above algorithms, you need to specify the following parameters:

Algorithm	Parameter	Parameter Description
C1-INVRQIMEN	Invoice Request Type for Individual Membership New Enrollment	<p>Used to indicate the invoice request type using which you want the system to automatically create an invoice request when an individual membership is activated.</p> <p>Note: This parameter is mandatory.</p>
	Invoice Request Type for Individual Membership Reinstatement	<p>Used to indicate the invoice request type using which you want the system to automatically create an invoice request when an individual membership is reinstated.</p> <p>Note: This parameter is optional.</p>
	Termination Reason	<p>Used to specify the termination reason. The system will automatically create an invoice request while reinstating an individual membership which was terminated due to the specified reason.</p> <p>Points to Note:</p> <p>This parameter is valid only when you specify the value for the Invoice Request Type for Individual Membership Reinstatement parameter.</p> <p>This parameter is optional. If you do not specify the termination reason, the system will create the invoice request on the individual membership reinstatement irrespective of the termination reason. However, if you specify the termination reason, the system will create the invoice request on the individual membership reinstatement only when the individual membership was terminated due to the specified reason.</p> <p>You can specify maximum five comma-separated values for this parameter.</p> <p>You must specify a reason which is already defined for the Terminated status of the C1-IndMembership business object in the Status Reason screen.</p>

Algorithm	Parameter	Parameter Description
C1-INVRQIMTM	Invoice Request Type	<p>Used to indicate the invoice request type using which you want the system to automatically create an invoice request when an individual membership is terminated.</p> <div data-bbox="862 415 1419 457" style="border: 1px solid black; padding: 2px;"> <p>Note: This parameter is mandatory.</p> </div>
	Termination Reason	<p>Used to specify the termination reason. The system will automatically create an invoice request when the individual membership is terminated due to the specified reason.</p> <div data-bbox="862 659 1419 1423" style="border: 1px solid black; padding: 2px;"> <p>Points to Note:</p> <p>This parameter is optional. If you do not specify the termination reason, the system will create the invoice request on the individual membership termination irrespective of the termination reason. However, if you specify the termination reason, the system will create the invoice request on the individual membership termination only when the individual membership is terminated due to the specified reason.</p> <p>You can specify maximum five comma-separated values for this parameter.</p> <p>You must specify a reason which is already defined for the Terminated status of the C1-IndMembership business object in the Status Reason screen.</p> </div>
C1-INVRQIMCN	Invoice Request Type	<p>Used to indicate the invoice request type using which you want the system to automatically create an invoice request when an individual membership is canceled.</p> <div data-bbox="862 1629 1419 1671" style="border: 1px solid black; padding: 2px;"> <p>Note: This parameter is mandatory.</p> </div>

Algorithm	Parameter	Parameter Description
	Cancellation Reason	<p>Used to specify the cancellation reason. The system will automatically create an invoice request when the individual membership is canceled due to the specified reason.</p> <div style="border: 1px solid black; padding: 5px;"> <p>Points to Note:</p> <p>This parameter is optional. If you do not specify the cancellation reason, the system will create the invoice request on the individual membership cancellation irrespective of the cancellation reason. However, if you specify the cancellation reason, the system will create the invoice request on the individual membership cancellation only when the individual membership is canceled due to the specified reason.</p> <p>You can specify maximum five comma-separated values for this parameter.</p> <p>You must specify a reason which is already defined for the Canceled status of the C1-IndMembership business object in the Status Reason screen.</p> </div>

The system determines the account to which the individual membership should be billed using the Account Identifier Type and Account Identifier Value characteristics which are defined on the individual membership. While fetching the characteristics, the system considers the characteristic types which are specified in the **Account Identifier Type Char Type** and **Account Identifier Value Char Type** option types of the **C1-ASOBLNG** feature configuration. Before automatically creating an invoice request for the account of an individual membership, the system validates whether the individual membership is eligible for the invoice request creation. If the individual membership is eligible for the invoice request creation, the system checks whether any invoice request already exists for the account in the non-final status. If an invoice request exists for the account in the non-final status, a new invoice request is not created for the account. However, if an invoice request does not exist for the account in the non-final status, a new invoice request is created for the account in the **Draft** status. The system then transitions the status of the invoice request from **Draft** to **Defer Processing Batch**.

Points to Note:

The system validates whether the individual membership is eligible for the invoice request creation only while activating an individual membership and not while terminating, reinstating, or canceling an individual membership.

The system enables you to cancel an invoice request which is in the **Defer Processing Batch** status.

When you execute the **BILLOPEN** batch with the **Off Cycle Switch** parameter set to **Y**, the system checks whether the account that meets the search criteria is included in any invoice request which is in the **Defer Processing Batch** status. If the account is included in the invoice request, the system checks whether the processing date of the invoice request is earlier than or equal to the batch business date. If the processing date is earlier than or equal to the batch business date, the system checks whether the pending bill already exists for the account. If the pending bill does not exist for the account, the system checks whether the cutoff date is later than the bill after date. If cutoff date is later than the bill after date, the system checks whether the accounting calendar is open for the accounting date. If the accounting calendar is open for the accounting date, the system creates the adhoc regular bill for the account in the **Pending** status. When you execute the **BSGENREG** batch with the **Off Cycle Switch** parameter set to **Y**, the system creates the bill segments for the adhoc regular bill. When you execute the **POSTPROC** batch with the **Off Cycle Switch** parameter set to **Y**, the system creates the post-processing bill segments (if any) and completes the adhoc regular bill of the account. Finally, the status of the invoice request is changed to **Processed**.

However, if the pending bill exists for the account, or if the cutoff date is earlier than or equal to the bill after date, or if the accounting calendar is not open for the accounting date, or if an error occurs while creating adhoc regular bill or its bill segments, or if a To Do is generated for the bill review, the status of the invoice request is changed to **Error**. The system enables you to move such erroneous invoice request back to the **Draft** status.

Collection Class

To introduce a delinquency process at different levels (i.e. person or account) in a person's hierarchy, the following three collection methods are available while defining a collection class:

- **Self-Control Delinquency** – Used when you want to create a distinct delinquency process for a person or an account that belongs to the collection class. You can use this collection method for both the fully insured group business and the fully insured individual business.
- **Parental Delinquency** – Used when you want to evaluate the debt of a person or an account that belongs to the collection class while monitoring the parent person's delinquency. You can use this collection method only for the fully insured group business.
- **Not Eligible for Collection** – Used when you to ignore a person or an account that belongs to the collection class while monitoring the delinquency. You can use this collection method for both the fully insured group business and the fully insured individual business.

Let us understand this with the help of an example.

Entity Type	Entity	Collection Class	Collection Method	System Behavior...
Account	A1	CC1	Self-Control Delinquency	<ul style="list-style-type: none"> The system creates a distinct delinquency process for the A1 account. The bills of the A1 account are considered while monitoring the delinquency of the A1 account and not while monitoring the delinquency of the account's main customer or any of its parent person in the hierarchy.
	A2	CC2	Parental Delinquency	<ul style="list-style-type: none"> The system does not create a distinct delinquency process for the A2 account. The bills of the A2 account are considered while monitoring the delinquency of the account's main customer or any of its parent person in the hierarchy.
	A3	CC3	Not Eligible for Collection	<ul style="list-style-type: none"> The system does not monitor the delinquency of the A3 account. The bills of the A3 account are not considered while monitoring the delinquency of the account's main customer or any of its parent person in the hierarchy.
Person	P1	CC4	Self-Control Delinquency	<ul style="list-style-type: none"> The system creates a distinct delinquency process for the P1 person. The bills of the P1's accounts and its child persons' accounts are considered while monitoring the delinquency of the P1 person and not while monitoring the delinquency of any of its parent person in the hierarchy.

Entity Type	Entity	Collection Class	Collection Method	System Behavior...
	P2	CC5	Parental Delinquency	<ul style="list-style-type: none"> If the P2 person has a parent person, the system does not create a distinct delinquency process for the P2 person. The bills of the P2's accounts and its child persons' accounts are considered while monitoring the delinquency of any of its parent person in the hierarchy. If the P2 person does not have a parent person, the system creates a distinct delinquency process for the P2 person. The bills of the P2's accounts and its child persons' accounts are considered while monitoring the delinquency of the P2 person.
	P3	CC6	Not Eligible for Collection	<ul style="list-style-type: none"> The system does not monitor the delinquency of the P3 person. But the system monitors the delinquency of the P3's accounts and its child person's when the collection method in the respective collection class is set to Self-Control Delinquency.

Delinquency Management

Most often, the fully insured group and individual businesses come across scenarios wherein customers fail to pay premium for their policy or health plans, respectively, on time, resulting in delinquency. For the fully insured group business, ORMB enables you to establish a delinquency process at the person (i.e. parent customer or bill group) level to whom the policies are billed or at the account (i.e. parent customer's or bill group's account) level to which the policies are billed. However, for the fully insured individual business, ORMB enables you to establish a delinquency process only at the account level to which individual memberships are billed. A delinquency process signifies a set of procedures and policies that are in place to manage overdue or unpaid insurance premiums.

The system refers the collection class of the person or account, respectively, to control its collection lifecycle. The system monitors the account's debt using the bills of the account as the collection objects. However, the system monitors the person's debt using the bills of all its accounts in its hierarchy as the collection objects. The following screens are introduced for the Delinquency Management feature:

- Delinquency Control
- Delinquency Event Type
- Delinquency Process Type

- Delinquency Process Search
- Delinquency Process

The system can create a delinquency process only when the following prerequisites are configured:

- If you want to create a delinquency process for a person or an account, you need to set the collection method of the respective collection class to **Self-Control Delinquency**. If you want to evaluate the debt of a person or an account while monitoring the parent person's delinquency, you need to set the collection method of the respective collection class to **Parental Delinquency**. However, if you want to ignore a person or an account while monitoring the delinquency, you need to set the collection method of the respective collection class to **Not Eligible for Collection**. For more information, refer to the [Collection Class](#) section.
- A delinquency control is defined for the required collection classes. For more information, refer to the [Delinquency Control](#) section.
- The required delinquency event types and delinquency process types are defined in the system. For more information about the delinquency event type and delinquency process type, refer to the [Delinquency Event Type](#) and [Delinquency Process Type](#) sections, respectively.
- A preference of the following preference categories is defined in the system:
 - [Delinquency Process](#)
 - [Delinquency Process Type](#)
 - [Routing Method – Contact Method Mapping](#)
- The parameters required to create the business rule criteria are defined in the system.
- The business rules of the following categories are defined in the system:
 - [Delinquency Event Attributes Business Rule](#)
 - [Delinquency Grace Period Business Rule](#)
 - [Delinquency Miscellaneous Options Business Rule](#)
 - [Delinquency Termination Date Rule Business Rule](#)
- Required rule types and rules are defined to determine whether the delinquency event is eligible to be included in the delinquency process of a person or an account. For more information, refer to the [Delinquency Event Eligibility Evaluation and Trigger Date Calculation](#) section.
- The outbound message types are defined for the following business objects:
 - C1-MemberTermRequest (required only for the fully insured individual business)
 - C1-MemberCancRequest (required only for the fully insured individual business)
 - C1-PolicyTerminateRequest (required only for the fully insured group business)
 - C1-PolicyReinstateRequest (required only for the fully insured group business)
- The following option types are defined in the **C1-ASOBLNG** feature configuration:
 - Individual Membership Configurations (required only for the fully insured individual business)
 - Membership Active Status (required only for the fully insured individual business)
 - Membership Terminated Status (required only for the fully insured individual business)
 - Policy Active Status (required only for the fully insured group business)

- Policy Pending Reinstatement Status (required only for the fully insured group business)
- Policy Pending Termination Status (required only for the fully insured group business)

For more information about these option types, refer to the [C1-ASOBLNG Feature Configuration](#) section.

- Set the batch control type of the following batches to **Timed** and execute them in the below given sequence:
 1. C1-DPMON
 2. C1-PRDLQ (required only for the fully insured group business)
 3. C1-ACDLQ
 4. C1-DPEVL
 5. C1-MBRNT (required only for the fully insured group business)
 6. C1-HLMON
 7. C1-ADMON (required only if the advance deposit feature is opted for the fully insured group business)
 8. C1-MTOMT (required only for the fully insured individual business)

Group Health Insurance Delinquency

The system enables you to create a delinquency process at the person or account level for the fully insured group business. You can create a delinquency process at the person or account level only when the collection method in the person's or account's collection class is set to **Self-Control Delinquency**.

In the account level delinquency process, the system monitors the debt of the account at predefined system events and commences the delinquency process whenever the account's debt is overdue. The system creates a delinquency process for an account only when at least one bill of the account is overdue. Note that the system ignores the account while monitoring the delinquency when the collection method in the respective collection class is set to **Not Eligible for Collection**.

However, in the person level delinquency process, the system considers all its accounts and its child persons' accounts where the collection method in their respective collection class is set to **Parental Delinquency**. It then monitors the debt of the person's and its child persons' accounts at predefined system events and commences the delinquency process whenever any of its account's debt is overdue. Note that, in the person's hierarchy, if the collection method in the collection class of any account is set to **Self-Control Delinquency**, then such account is not considered while monitoring the parent person's delinquency. The system creates a distinct delinquency process for each such account where the collection method in the respective collection class is set to **Self-Control Delinquency**. Similarly, in the person's hierarchy, if the collection method in the collection class of any child person is set to **Self-Control Delinquency**, then all the child person's accounts and its grandchild persons' accounts are not considered in the parent person's delinquency. The system creates a distinct delinquency process for each such person (i.e. bill group) where the collection method in the respective collection class is set to **Self-Control Delinquency**. The system creates a delinquency process for a person only when at least one bill of any account (considered) in its hierarchy is overdue.

Note that the system ignores the persons and accounts in the person's hierarchy while monitoring the delinquency when the collection method in the respective collection class is set to **Not Eligible for Collection**.

Individual Health Insurance Delinquency

The system enables you to create a delinquency process at the account level for the fully insured individual business. You can create a delinquency process at the account level only when the collection method in the respective collection class is set to **Self-Control Delinquency**. In the account level delinquency process, the system monitors the debt of the account at predefined system events and commences the delinquency process whenever the account's debt is overdue.

Note that the system ignores the account while monitoring the delinquency when the collection method in the respective collection class is set to **Not Eligible for Collection**.

Account Level Delinquency Process

To create a delinquency process at the account level, the system requires the following collection details of the account:

- **Collection Class** – Used to control how the account's debt is compared against collection criteria (i.e. tolerance limit) to determine whether a collection process (such as delinquency process) should be initiated for the account. A new collection method named **Self-Control Delinquency** is available while defining a collection class. It is used to create a distinct or separate delinquency process for the accounts which belong to the collection class. You can specify the collection class for an account from the user interface or through a health care inbound message.

Points to Note:

If you do not want to initiate delinquency process for an account, ensure that the account belongs to a collection class where the collection method is set to **Not Eligible for Collection**.

If the collection class is not specified for an account, the system, by default, sets it to the one specified in the respective customer class when you create the account through a health care inbound message.

- **Credit Review Grace Days** – Used to indicate the number of days from the bill due date or the processing date after which the account's debt should be monitored for delinquency. For example, if the bill due date is 07-Jan-2024 and credit review grace days is set to 10, then the system will consider the account for monitoring the delinquency on or after 17-Jan-2024. You can specify the credit review grace days for the customer class to which the account belongs from the user interface.

Note: The system considers the bill due date in case of bill events and the processing date in case of non-bill events while monitoring the delinquency of an account.

- **Next Credit Review Date** – Used to indicate the date when the account should be considered for monitoring the delinquency. The system calculates the next credit review date for the account using the bill due date or the processing date and the credit review grace days. The next credit review date of the account is stored in the **CI_ADM_RVW_SCH** table. The system calculates the next credit review date for the account on the following system events:
 - Bill Completion
 - Payment Cancellation (due to Non-Sufficient Funds)
 - Active Overdue Process Review
 - Match Event is Added, Changed, or Deleted
 - Payment Freeze
 - Payment Agreement Request or Promise to Pay is Broken or Canceled
 - Written Off Bill is Reversed
- **Postpone Credit Review Until** – Used when you want to review the account’s debt after a particular date. The system will consider the account for monitoring the delinquency only after the specified date. You can specify the postpone credit review until date for an account from the user interface or through a health care inbound message.
- **Credit Rating** – Used to indicate the credit rating of the account. It is calculated by the system using the credit rating transactions which are effective on the system date. The system then considers the sum of account’s credit rating and credit rating threshold (defined in the installation options) to determine whether the account’s debt should be monitored for delinquency.
- **Minimum Credit Review Frequency** – Used to indicate the frequency at which the accounts belonging to the respective collection class should be monitored for delinquency. In other words, it indicates in how many days the system should review whether the bills of the accounts are overdue. You can specify the minimum credit review frequency for a delinquency control of a collection class from the user interface.
- **Last Credit Review Date** – Used to indicate the date when the account's debt was last reviewed. The system stamps the last credit review date against an account when a delinquency process is initiated for the account through the **C1-ACDLQ** batch.
- **Drag Days** – Used to specify additional grace days (if any) offered at the account level. These drag days are considered while calculating the grace end date when the trigger date calculation mode (in the respective delinquency process type) is set to **Latest Bill Due Date with Grace Period**. For example, if the latest bill due date is 05-Jan-2024, grace period is set to 5 days (in the respective delinquency process type), and drag days is to 10 days (for the respective account), then the grace end date for the account is set to 20-Jan-2024.

Account Level Delinquency Process Creation

A new batch named **C1-ACDLQ** is introduced in this release. It is used to create a delinquency process for an account when its overdue bills do not meet the tolerance limit. This batch considers the accounts which meet the following conditions:

- Collection method of the collection class to which the account belongs is set to **Self-Control Delinquency**

- Next credit review date is either not calculated for the account or is earlier than or equal to the batch business date
- Postpone credit review until date is either not defined for the account or is earlier than or equal to the batch business date
- The date derived using the last credit review date + minimum credit review frequency (defined in the delinquency control) is either blank or is earlier than or equal to the batch business date

Note: The next credit review date of an account is stored in the **CI_ADM_RVW_SCH** table and the postpone credit review until date of an account is stored in the **CI_ACCT** table.

Once the list of accounts are derived, the system executes the delinquency monitor rule algorithm which is attached to the delinquency control of the respective account's collection class. If the system is not able to derive the delinquency monitor rule algorithm for an account, an appropriate error message is shown in the batch run tree. If the system is able to derive the delinquency monitor rule algorithm for an account and if the new delinquency process is successfully created for an account or if an existing delinquency process is updated for an account, this batch sets the last credit review date of the account in the **CI_ACCT** table to the processing date.

For more information on how the **C1-DLNQMNR** algorithm (attached to the delinquency control) derives the overdue bills, refer to the [Delinquency Control](#) section.

This batch is a multi-threaded batch. The multi-threading is based on account ID and chunks for multi-threading are created based on numerical distribution of account ID. This batch contains the following parameters:

Parameter Name	Mandatory (Yes or No)	Description
Division	Yes (Conditional)	Used when you want to monitor the delinquency of the accounts which belong to a particular division. Note: This parameter is required when you specify a customer class while executing this batch.
Customer Class	No	Used when you want to monitor the delinquency of the accounts which belong to a particular customer class.
Chunk Size	Yes	Used to specify the number of accounts you want to process in each work unit. Note: By default, the value is set to 5.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

If a new delinquency process is successfully created for an account, the status of the delinquency process is set to the value (for example, **Initiated**) specified in the **Delinquency Process Initiated Status** attribute of the delinquency process preference. Note that the system considers the preference which is specified in the **Delinquency Process Field Mapping** option type of the **DELINPROC** feature configuration. For more information about the feature configuration, refer to the [DELINPROC Feature Configuration](#) section.

The system then invokes the **C1-DPEVTCR** algorithm which is attached to the **Initiated** status. The **C1-DPEVTCR** algorithm creates a delinquency event of each delinquency event type that is added in the delinquency process type using which the delinquency process is created. If the delinquency event trigger mode is set to **Manual** in the delinquency process type, the status of the delinquency event is set to **Pending**. However, if the delinquency event trigger mode is set to **Automatic or Manual** or **Automatic** in the delinquency process type, the status of the delinquency event is set to **Pending Evaluation**.

Once the delinquency events are created for the delinquency process, this algorithm does the following:

- Sets the delinquency due date of the delinquency process to the latest bill due date
- Derives the grace period from the delinquency process type or using an algorithm attached to the **Grace Period Derivation** system event depending on the grace period source which is specified in the delinquency process type
- Sets the grace start date in the following manner:

Trigger Date Calculation Mode	Usage Basis	Grace Start Date Basis	Grace Start Date is set to...
Latest Bill Due Date with Grace Period	-	-	Latest Bill Due Date
Usage Basis	Delinquency Process Creation Date	-	Delinquency Process Creation Date
Usage Basis	Grace Start Date	Latest Bill Due Date	Latest Bill Due Date
Usage Basis	Grace Start Date	X Days after Latest Bill Due Date	Latest Bill Due Date + Add Days To Due Date
Usage Basis	Grace Start Date	Delinquency Process Creation Date	Delinquency Process Creation Date
Usage Basis	Latest Bill Date	-	Latest Bill Date
Usage Basis	Latest Bill Due Date	-	Latest Bill Due Date

- Calculates the grace end date in the following manner:

Trigger Date Calculation Mode	Usage Basis	Grace Start Date Basis	Grace Start Date	Grace End Date is set to...
Latest Bill Due Date with Grace Period	-	-	Latest Bill Due Date	Grace Start Date + Grace Period + Drag Days (if any, defined at the respective level (i.e. account or person))
Usage Basis	Delinquency Process Creation Date	-	Delinquency Process Creation Date	Grace Start Date + Grace Period
Usage Basis	Grace Start Date	Latest Bill Due Date	Latest Bill Due Date	Grace Start Date + Grace Period
Usage Basis	Grace Start Date	X Days after Latest Bill Due Date	Latest Bill Due Date + Add Days To Due Date	Grace Start Date + Grace Period
Usage Basis	Grace Start Date	Delinquency Process Creation Date	Delinquency Process Creation Date	Grace Start Date + Grace Period
Usage Basis	Latest Bill Date	-	Latest Bill Date	Grace Start Date + Grace Period
Usage Basis	Latest Bill Due Date	-	Latest Bill Due Date	Grace Start Date + Grace Period

Note: This algorithm does not calculate the trigger date and preview date for the events when delinquency event trigger mode is set to **Manual** in the delinquency process type. However, the system calculates the trigger date and preview date for the events when delinquency event trigger mode is set to **Automatic or Manual** or **Automatic** in the delinquency process type while executing the **C1-DPEVL** batch.

Once a delinquency process is created for an account, you can perform the following actions on the delinquency process from the user interface:

- Add additional information in the form of comments or characteristics
- Manually hold a delinquency process and edit the hold details (if required)
- Manually release a hold on the delinquency process
- Manually cancel a delinquency process
- Edit the trigger date of the delinquency events in the delinquency process
- Trigger or skip the delinquency events in the delinquency process
- Request to reinstate the delinquency process (only applicable for the fully insured group business)
- Approve, reject, or withdraw an approval record of the delinquency process

For more information about the manual actions, refer to the [Delinquency Process Maintenance](#) section. The system also enables you to do the following:

- Automatically hold a delinquency process when the certain conditions are met
- Automatically release a hold on the delinquency process when the certain conditions are met
- Automatically cancel a delinquency process when the certain conditions are met
- Automatically resume a cancelled delinquency process when the certain conditions are met
- Automatically reinstate a delinquency process when the certain conditions are met (only applicable for the fully insured group business)

To enable the automatic process for the above actions, you need to accordingly configure the delinquency process type. The system also enables you to optionally configure the approval process for various manual actions that you can perform on the delinquency event or delinquency process. For more information, refer to the [Delinquency Process Approval](#) section.

A delinquency event and process goes through various statuses in its lifecycle. The delinquency process status also accordingly changes when the policies or individual memberships billed to the account are terminated or reinstated.

Person Level Delinquency Process

To create a delinquency process at the person level, the system requires the following collection details of the person:

- **Collection Class** – Used to control how the person’s debt is compared against collection criteria (i.e. tolerance limit) to determine whether a collection process (such as delinquency process) should be initiated for the person. A new collection method named **Self-Control Delinquency** is available while defining a collection class. It is used to create a distinct or separate delinquency process for the persons who belong to the collection class. You can specify the collection class for a person from the user interface or through a health care inbound message.

If you want to evaluate the debt of the person’s accounts, its child persons, and the child persons’ accounts while monitoring the person’s delinquency, you must set the collection method of their respective collection class to **Parental Delinquency**. If you set the collection method of the collection class for any of the person’s account, any of its child person, or child persons’ accounts to **Self-Control Delinquency**, the system does not consider the debt of the respective entity while monitoring the delinquency of the parent person.

Points to Note:

If you do not want to initiate delinquency process for a child person or an account, ensure that the person or account belongs to a collection class where the collection method is set to **Not Eligible for Collection**.

If the collection class is not defined for a person, the system skips the person while monitoring the delinquency.

- **Next Credit Review Date** – Used to indicate the date when the person should be considered for monitoring the delinquency. The system calculates the next credit review date for the account on the following system events:
 - Bill Completion
 - Payment Cancellation (due to Non-Sufficient Funds)

- Active Overdue Process Review
- Match Event is Added, Changed, or Deleted
- Payment Freeze
- Payment Agreement Request or Promise to Pay is Broken or Canceled
- Written Off Bill is Reversed

If the collection method in the account's collection class is set to **Parental Delinquency**, the system automatically traverses up the hierarchy and checks whether the collection method in the collection class of its main customer or any of its parent person in the hierarchy is set to **Self-Control Delinquency**. If so, the system sets the next credit review date of the person to the account's next credit review date. The next credit review date of the person is stored in the **C1_PER_RVW_SCH** table.

- **Postpone Credit Review Until** – Used when you want to review the person's debt after a particular date. The system will consider the person for monitoring the delinquency only after the specified date. You can specify the postpone credit review until date for a person through a health care inbound message.
- **Credit Rating** – Used to indicate the credit rating of the person. It is calculated by adding the credit rating of all the accounts where the person is the main customer. The system then considers the sum of person's credit rating and credit rating threshold (defined in the installation options) to determine whether determine whether the person's debt should be monitored for delinquency.
- **Minimum Credit Review Frequency** – Used to indicate the frequency at which the persons belonging to the respective collection class should be monitored for delinquency. In other words, it indicates in how many days the system should review whether the bills of the persons are overdue. You can specify the minimum credit review frequency for a delinquency control of a collection class from the user interface.
- **Last Credit Review Date** – Used to indicate the date when the person's debt was last reviewed. The system stamps the last credit review date against a person when a delinquency process is initiated for the person through the **C1-PRDLQ** batch.
- **Drag Days** - Used to specify additional grace days (if any) offered at the person level. These drag days are considered while calculating the grace end date when the trigger date calculation mode (in the respective delinquency process type) is set to **Latest Bill Due Date with Grace Period**. For example, if the latest bill due date is 05-Jan-2024, grace period is set to 5 days (in the respective delinquency process type), and drag days is to 10 days (for the respective person), then the grace end date for the person is set to 20-Jan-2024.

Person Level Delinquency Process Creation

A new batch named **C1-PRDLQ** is introduced in this release. It is used to create a delinquency process for a person when its overdue bills do not meet the tolerance limit. This batch considers the persons which meet the following conditions:

- Collection method of the collection class to which the person belongs is set to **Self-Control Delinquency**

- Next credit review date is either not derived for the person or is earlier than or equal to the batch business date
- Postpone credit review until date is either not defined for the person or is earlier than or equal to the batch business date
- The date derived using the last credit review date + minimum credit review frequency (defined in the delinquency control) is either blank or is earlier than or equal to the batch business date

Note: The next credit review date of a person is stored in the **C1_PER_RVW_SCH** table and the postpone credit review until date of a person is stored in the **CI_PER** table.

Once the list of persons are derived, the system executes the delinquency monitor rule algorithm which is attached to the delinquency control of the respective person's collection class. If the system is not able to derive the delinquency monitor rule algorithm for a person, an appropriate error message is shown in the batch run tree. If the system is able to derive the delinquency monitor rule algorithm for a person and if the new delinquency process is successfully created for a person or if an existing delinquency process is updated for a person, this batch sets the last credit review date of the person in the **CI_PER** table to the processing date.

For more information on how the **C1-DLNQMNR** algorithm (attached to the delinquency control) derives the overdue bills, refer to the [Delinquency Control](#) section.

This batch is a multi-threaded batch. The multi-threading is based on person ID and chunks for multi-threading are created based on numerical distribution of person ID. This batch contains the following parameters:

Parameter Name	Mandatory (Yes or No)	Description
Division	Yes (Conditional)	Used when you want to monitor the delinquency of the persons which belong to a particular division. Note: This parameter is required when you specify a customer class while executing this batch.
Customer Class	No	Used when you want to monitor the delinquency of the persons which belong to a particular customer class.
Chunk Size	Yes	Used to specify the number of persons you want to process in each work unit. Note: By default, the value is set to 0.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

If a new delinquency process is successfully created for a person, the status of the delinquency process is set to the value (for example, **Initiated**) specified in the **Delinquency Process Initiated Status** attribute of the delinquency process preference. Note that the system considers the preference which is specified in the **Delinquency Process Field Mapping** option type of the **DELINPROC** feature configuration. For more information about the feature configuration, refer to the [DELINPROC Feature Configuration](#) section.

The system then invokes the **C1-DPEVTCR** algorithm which is attached to the **Initiated** status. The **C1-DPEVTCR** algorithm creates a delinquency event of each delinquency event type that is added in the delinquency process type using which the delinquency process is created. If the delinquency event trigger mode is set to **Manual** in the delinquency process type, the status of the delinquency event is set to **Pending**. However, if the delinquency event trigger mode is set to **Automatic or Manual** or **Automatic** in the delinquency process type, the status of the delinquency event is set to **Pending Evaluation**.

Once the delinquency events are created for the delinquency process, this algorithm does the following:

- Sets the delinquency due date of the delinquency process to the latest bill due date
- Derives the grace period from the delinquency process type or using an algorithm attached to the **Grace Period Derivation** system event depending on the grace period source which is specified in the delinquency process type
- Sets the grace start date in the following manner:

Trigger Date Calculation Mode	Usage Basis	Grace Start Date Basis	Grace Start Date is set to...
Latest Bill Due Date with Grace Period	-	-	Latest Bill Due Date
Usage Basis	Delinquency Process Creation Date	-	Delinquency Process Creation Date
Usage Basis	Grace Start Date	Latest Bill Due Date	Latest Bill Due Date
Usage Basis	Grace Start Date	X Days after Latest Bill Due Date	Latest Bill Due Date + Add Days To Due Date
Usage Basis	Grace Start Date	Delinquency Process Creation Date	Delinquency Process Creation Date
Usage Basis	Latest Bill Date	-	Latest Bill Date
Usage Basis	Latest Bill Due Date	-	Latest Bill Due Date

- Calculates the grace end date in the following manner:

Trigger Date Calculation Mode	Usage Basis	Grace Start Date Basis	Grace Start Date	Grace End Date is set to...
Latest Bill Due Date with Grace Period	-	-	Latest Bill Due Date	Grace Start Date + Grace Period + Drag Days (if any, defined at the respective level (i.e. account or person))
Usage Basis	Delinquency Process Creation Date	-	Delinquency Process Creation Date	Grace Start Date + Grace Period
Usage Basis	Grace Start Date	Latest Bill Due Date	Latest Bill Due Date	Grace Start Date + Grace Period
Usage Basis	Grace Start Date	X Days after Latest Bill Due Date	Latest Bill Due Date + Add Days To Due Date	Grace Start Date + Grace Period
Usage Basis	Grace Start Date	Delinquency Process Creation Date	Delinquency Process Creation Date	Grace Start Date + Grace Period
Usage Basis	Latest Bill Date	-	Latest Bill Date	Grace Start Date + Grace Period
Usage Basis	Latest Bill Due Date	-	Latest Bill Due Date	Grace Start Date + Grace Period

Note: This algorithm does not calculate the trigger date and preview date for the events when delinquency event trigger mode is set to **Manual** in the delinquency process type. However, the system calculates the trigger date and preview date for the events when delinquency event trigger mode is set to **Automatic or Manual** or **Automatic** in the delinquency process type while executing the **C1-DPEVL** batch.

Once a delinquency process is created for a person, you can perform the following actions on the delinquency process from the user interface:

- Add additional information in the form of comments or characteristics
- Manually hold a delinquency process and edit the hold details (if required)
- Manually release a hold on the delinquency process
- Manually cancel a delinquency process
- Edit the trigger date of the delinquency events in the delinquency process
- Trigger or skip the delinquency events in the delinquency process
- Request to reinstate the delinquency process (only applicable for the fully insured group business)

For more information about the manual actions, refer to the [Delinquency Process Maintenance](#) section. The system also enables you to do the following:

- Automatically hold a delinquency process when the certain conditions are met

- Automatically release a hold on the delinquency process when the certain conditions are met
- Automatically cancel a delinquency process when the certain conditions are met
- Automatically resume a cancelled delinquency process when the certain conditions are met
- Automatically reinstate a delinquency process when the certain conditions are met (only applicable for the fully insured group business)

To enable the automatic process for the above actions, you need to accordingly configure the delinquency process type. The system also enables you to optionally configure the approval process for various manual actions that you can perform on the delinquency event or delinquency process. For more information, refer to the [Delinquency Process Approval](#) section.

A delinquency event and process goes through various statuses in its lifecycle. The delinquency process status also accordingly changes when the policies or individual memberships billed to the person are terminated or reinstated.

Types of Delinquency Processes

You can design different types of delinquency processes based on the fully insured group and fully insured individual business requirements. For example,

- Some may be used to just send warning or harsh letters and/or notifications. Such delinquency processes are applicable to both the fully insured group and fully insured individual businesses.
- Some may be used to initiate the termination request for the individual memberships which are billed to a delinquent account. Such delinquency processes are applicable only for the fully insured individual business.
- Some may be used to initiate the cancellation request for the individual memberships which are billed to a delinquent account. Such delinquency processes are applicable only for the fully insured individual business.
- Some may be used to initiate the termination request for the policies which are billed to a delinquent customer or account. Such delinquency processes are applicable only for the fully insured group business.
- Some may be used to initiate the reinstatement request for the policies which are billed to a delinquent customer or account. Such delinquency processes are applicable only for the fully insured group business.
- Some may be used to settle the unpaid dues for a delinquent customer or account using the advance deposit maintained at the person or policy level. Such delinquency processes are applicable only for the fully insured group business.

The system enables you to send different warning letters or To Do notifications, initiate the termination or cancellation request for the individual memberships, initiate the termination or reinstatement request for the policies, or apply advance deposit to settle the unpaid dues by triggering different delinquency events at predefined intervals through a different delinquency process. The system creates a delinquency process for a person or an account using a delinquency process type. The system determines the delinquency process type using which the delinquency process should be created for a person or an account through the delinquency control. Note that the system considers the delinquency control that is defined for the collection class to which the person or account belongs. For more information about the delinquency control and delinquency process type, refer to the [Delinquency Control](#) and [Delinquency Process Type](#) sections, respectively.

The system enables you to configure the events that should be triggered in a delinquency process by adding appropriate delinquency event types in a delinquency process type. For more information about the delinquency event type, refer to the [Delinquency Event Type](#) section. The different types of events are triggered in different types of delinquency processes. The system enables you to design the following types of delinquency processes for different set of customers:

- [Debt Notification Delinquency Process](#) (applicable to both group and individual customers)
- [Debt Collection Letters Delinquency Process](#) (applicable to both group and individual customers)
- [Membership Cancellation Delinquency Process](#) (applicable to only individual customers)
- [Membership Termination Delinquency Process](#) (applicable to only individual customers)
- [Advance Deposit Delinquency Process](#) (applicable to only group customers)

Debt Notification Delinquency Process

There might be scenarios wherein the group or individual customers fail to pay premium for their policy plan or health plan coverage, respectively. In such cases, you may want to send reminder notifications to the group or individual customers to pay their premium immediately without any delay. Oracle Revenue Management and Billing enables you to send one or more To Do notifications at regular intervals to:

- The person (i.e. parent customer or bill group) when the delinquency process is created for a person
- The main customer of the account when the delinquency process is created for an account

To create a delinquency process to send To Do notifications, you need to design the delinquency process type (using which you want to create the delinquency process) in the following manner:

Delinquency Process Type	Delinquency Type	Event	System Event	Algorithm	Algorithm Purpose
DP1	Send First Reminder Notification		Event Activation	C1-DLQCRTODO1	Creates a To Do using the given To Do type
	Send Second Reminder Notification		Event Activation	C1-DLQCRTODO2	Creates a To Do using the given To Do type
	Send Third Reminder Notification		Event Activation	C1-DLQCRTODO3	Creates a To Do using the given To Do type

Note that the **C1-DLQCRTODO** algorithm is shipped with the product. However, you can create different algorithms (for example, **C1-DLQCRTODO1**, **C1-DLQCRTODO2**, and **C1-DLQCRTODO3**) using the **C1-DLQCRTODO** algorithm type when you want to send different reminder notifications at regular intervals to the person or the main customer of the account. For more information about the **C1-DLQCRTODO** algorithm, refer to the [Delinquency Event Type](#) section.

Note: If you want to send the same reminder notification (i.e. the same message) multiple times at different intervals, you can use the **C1-DLQCRTODO** algorithm in all the delinquency event types. However, if you want to send different reminder notifications (i.e. the different message) at different intervals, you can create different algorithms (for example, **C1-DLQCRTODO1**, **C1-DLQCRTODO2**, and **C1-DLQCRTODO3** as shown above) and attach it to the delinquency event type in the sequence in which you want to send the reminders.

Once a delinquency process is created for a person or an account to send To Do notifications, the status of the delinquency process is set to the value (for example, **Initiated**) specified in the **Delinquency Process Initiated Status** attribute of the delinquency process preference. Once the first event in the delinquency process is triggered, the status of the delinquency process is changed to the value (for example, **Delinquency In Progress**) specified in the **Delinquency Process InProgress Status** attribute of the delinquency process preference. Once a To Do is created for the person or the main customer of the account, the status of the delinquency event is changed to **Completed**. Once all the delinquency events are triggered successfully, the status of the delinquency process is changed to the value (for example, **Completed**) specified in the **Delinquency Process Completed Status** attribute of the delinquency process preference.

The system may receive a payment or an adjustment against the overdue bill for which a delinquency process is created to send To Do notifications. In such case, the system behaves in the following manner depending on the status of the delinquency process:

- If a payment or an adjustment is made when a delinquency process is in the **Initiated, Delinquency In Progress, On Hold** status, the status of the delinquency process is changed to the value (for example, **Canceled**) specified in the **Delinquency Process Canceled Status** attribute of the delinquency process preference. For more information on how the delinquency process is automatically canceled, refer to the [Canceling a Delinquency Process](#) section.
- If a payment or an adjustment is made when a delinquency process is in the **Pending Termination** status, the status of the delinquency process is changed to the value (for example, **Canceled**) specified in the **Delinquency Process Canceled Status** attribute of the delinquency process preference. For more information on how the delinquency process is automatically canceled, refer to the [Canceling a Delinquency Process](#) section. This scenario is valid only for the fully insured individual business. This scenario is addressed only when such delinquency processes are automatically canceled through the **C1-DPMON** batch.
- If a payment or an adjustment is made when a delinquency process is in the **Completed** status, no action takes place on the debt notification delinquency process.

However, if the payment or adjustment is canceled due to any reason, the system executes the **C1-CDPPAYCAN** or **C1-DPADJCAN** algorithm attached to the customer class or adjustment type, respectively, to resume the delinquency process. For more information on how the delinquency process is automatically resumed, refer to the [Resuming a Delinquency Process](#) section.

Points to Note:

If a payment or an adjustment is canceled when a debt notification delinquency process is in the **Completed** status, no action takes place on the debt notification delinquency process.

The system considers the delinquency process preference which is specified in the **Delinquency Process Field Mapping** option type of the **DELINPROC** feature configuration. For more information about the feature configuration, refer to the [DELINPROC Feature Configuration](#) section.

Debt Collection Letters Delinquency Process

There might be scenarios wherein the group or individual customers fail to pay premium for their policy plan or health plan coverage, respectively, even after sending continuous reminders through To Do notifications. In such scenarios, you may want to send them dunning or warning letters (i.e. delinquency notices). Oracle Revenue Management and Billing enables you to send one or more warning or dunning letters via customer contact to:

- Parent Customer, Bill Group, and/or Billing accounts when the delinquency process is created for a person
- All the persons (that can receive notification) associated with the account when the delinquency process is created for an account

To create a delinquency process to send warning or dunning letters, you need to design the delinquency process type (using which you want to create the delinquency process) in either of the following manner:

Delinquency Process Type	Delinquency Event Type	System Event	Algorithm	Algorithm Purpose
DP1	Send First Reminder Notification	Event Activation	C1-DLQCRTODO	Creates a To Do using the given To Do type
	Send a Warning Letter	Event Activation	C1-DLQSENDCC1	Sends a warning letter via a customer contact
		Monitor Completed Event	C1-RCLTRGDMD	Recalculates trigger dates for remaining events based on document mail date
	Send a Dunning Letter	Event Activation	C1-DLQSENDCC2	Sends a dunning letter via a customer contact
DP2	Send a Warning Letter	Event Activation	C1-DLQSENDCC1	Sends a warning letter via a customer contact

Delinquency Process Type	Delinquency Event Type	System Event	Algorithm	Algorithm Purpose
	Recalculate Trigger Dates	Monitor Completed Event	C1-RCLTRGDMD	Recalculates trigger dates for remaining events based on document mail date
	Send a Dunning Letter	Event Activation	C1-DLQSENDCC2	Sends a dunning letter via a customer contact
DP3	Send a Warning Letter	Event Activation	C1-DLQSENDCC1	Sends a warning letter via a customer contact
	Send a Dunning Letter	Event Activation	C1-DLQSENDCC2	Sends a dunning letter via a customer contact

The **C1-DLQSENDCC** algorithm is shipped with the product. You can create different algorithms (for example, **C1-DLQSENDCC1** and **C1-DLQSENDCC2**) using the **C1-DLQSENDCC** algorithm type when you want to send different debt collection letters at different intervals. Note that sending a To Do notification before sending debt collection letters through a delinquency process is optional. You may or may not include these To Do notification event types while designing a delinquency process type to send debt collection letters. In addition, the **C1-RCLTRGDMD** algorithm attached to the **Monitor Completed Event** system event is optional. If required, you can recalculate the trigger dates for the remaining events in the delinquency process based on document mail date. For more information about the **C1-DLQSENDCC** and **C1-RCLTRGDMD** algorithms, refer to the [Delinquency Event Type](#) section.

Note: You can attach an algorithm to either or both of the system events (i.e. **Event Activation** and **Monitor Completed Event**) in a delinquency event type.

Once a delinquency process is created for a person or an account to send debt collection letters, the status of the delinquency process is set to the value (for example, **Initiated**) specified in the **Delinquency Process Initiated Status** attribute of the delinquency process preference. Once the first event in the delinquency process is triggered, the status of the delinquency process is changed to the value (for example, **Delinquency In Progress**) specified in the **Delinquency Process InProgress Status** attribute of the delinquency process preference. A customer contact may be created when the first or subsequent event is triggered in the delinquency process. For more information about the person to whom a customer contact is sent, refer to the **C1-DLQSENDCC** algorithm description in the [Delinquency Event Activation](#) section. If the member level notification needs to be sent to all the derived group or individual memberships, the system also creates a customer contact for the main subscriber of each membership. For more information about the member level notifications, refer to the [Creating Customer Contacts for Group or Individual Memberships](#) section. Once all the required customer contacts are created, the status of the delinquency event is changed to **Completed**. Once all the delinquency events are triggered successfully, the status of the delinquency process is changed to the value (for example, **Completed**) specified in the **Delinquency Process Completed Status** attribute of the delinquency process preference.

The system may receive a payment or an adjustment against the overdue bill for which a delinquency process is created to send debt collection letters. In such case, the system behaves in the following manner depending on the status of the delinquency process:

- If a payment or an adjustment is made when a delinquency process is in the **Initiated, Delinquency In Progress, or On Hold** status, the status of the delinquency process is changed to the value (for example, **Canceled**) specified in the **Delinquency Process Canceled Status** attribute of the delinquency process preference. For more information on how the delinquency process is automatically canceled, refer to the [Canceling a Delinquency Process](#) section.
- If a payment or an adjustment is made when a delinquency process is in the **Pending Termination** status, the status of the delinquency process is changed to the value (for example, **Canceled**) specified in the **Delinquency Process Canceled Status** attribute of the delinquency process preference. For more information on how the delinquency process is automatically canceled, refer to the [Canceling a Delinquency Process](#) section. This scenario is valid only for the fully insured individual business. This scenario is addressed only when such delinquency processes are automatically canceled through the **C1-DPMON** batch.
- If a payment or an adjustment is made when a delinquency process is in the **Completed** status, no action takes place on the debt collection letters delinquency process.

However, if the payment or adjustment is canceled due to any reason, the system executes the **C1-CDPPAYCAN** or **C1-DPADJCAN** algorithm attached to the customer class or adjustment type, respectively, to resume the delinquency process. For more information on how the delinquency process is automatically resumed, refer to the [Resuming a Delinquency Process](#) section.

Points to Note:

If a payment or an adjustment is canceled when a debt collection letters delinquency process is in the **Completed** status, no action takes place on the debt collection letters delinquency process.

The system considers the delinquency process preference which is specified in the **Delinquency Process Field Mapping** option type of the **DELINPROC** feature configuration. For more information about the feature configuration, refer to the [DELINPROC Feature Configuration](#) section.

Individual Membership Cancellation Delinquency Process

There might be scenarios wherein the individual customers fail to pay the binder payment for their health plan coverage. As a result, their individual memberships cannot be activated in the system. Such individual memberships should be canceled in the system in a timely manner. To adhere to this requirement, you can now configure the system such that a delinquency process is created for the individual membership's account to send a cancellation outbound message to the enrollment system when the payment for the first bill (i.e. binder payment) is not made within the due date. This enables you to initiate the cancellation process for an individual membership through the **Delinquency Management** feature. Only when an inbound message is received from the enrollment system to cancel an individual membership, the system changes the status of the individual membership to **Canceled**.

To create a delinquency process to initiate cancellation request for an individual membership through an outbound message, you need to design the delinquency process type (using which you want to create the delinquency process) in the following manner:

Delinquency Process Type	Delinquency Event Type	System Event	Algorithm	Algorithm Purpose
DP1	Send First Reminder Notification	Event Activation	C1-DLQCRTODO	Creates a To Do using the given To Do type
	Send a Warning Letter	Event Activation	C1-DLQSENDCC	Sends a warning letter via a customer contact
	Update Individual Membership Cancellation Reason	Event Activation	C1-DLEVMBNCNC	Updates the status reason of the individual memberships billed to the delinquent account

For more information about these algorithms, refer to the [Delinquency Event Type](#) section. Note that sending a To Do notification and a warning or dunning letter before initiating the cancellation request for an individual membership through a delinquency process is optional. You may or may not include these event types while designing a delinquency process type to initiate cancellation request for individual memberships billed to the delinquent account. However, you need to attach the **C1-DLEVMBNCNC** algorithm to initiate cancellation request for individual memberships billed to the delinquent account. The **C1-DLEVMBNCNC** algorithm updates the status reason of the individual memberships which are billed to the delinquent account. It considers the status reason which is specified in the **Awaiting Membership Cancellation Reason** attribute of the delinquency process preference. Before updating the status reason of the individual membership, the system validates whether the status reason is defined for the **Active** status of the **C1-IndMembership** business object.

Once a delinquency process is created for an account, the status of the delinquency process is set to the value (for example, **Initiated**) specified in the **Delinquency Process Initiated Status** attribute of the delinquency process preference. Once the first event in the delinquency process is triggered, the status of the delinquency process is changed to the value (for example, **Delinquency In Progress**) specified in the **Delinquency Process InProgress Status** attribute of the delinquency process preference. Once the status reason of the active individual memberships is updated, the status of the delinquency event is changed to **Complete**. If all the events in the delinquency process are triggered successfully, the status of the delinquency process is changed to the value (for example, **Completed**) specified in the **Delinquency Process Completed Status** attribute of the delinquency process preference. However, if there are any more events that are yet to be triggered, the status of the delinquency process remains as **Delinquency In Progress**.

You need to then execute the **C1-MTOMT** batch to generate the cancellation outbound messages for the individual memberships. For more information about the batch, refer to the [Outbound Message Generation for Delinquency Management](#) section. Before creating the cancellation outbound messages for the individual memberships, you need to setup some pre-requisites for the cancellation outbound message generation. For more information about the pre-requisites, refer to the [Membership Cancellation Outbound Message](#) section.

Once the cancellation outbound message is sent to the enrollment system, you need to wait for the enrollment system to send the inbound message to cancel the individual membership in ORMB. Once the inbound message to cancel an individual membership is processed, the status of the individual membership is changed to **Canceled**. Once the individual membership is canceled in ORMB, the status of the delinquency process is changed to the value (for example, **Canceled**) specified in the **Delinquency Process Canceled Status** attribute of the delinquency process preference when the **C1-DPMON** batch is executed.

The system may receive a payment or an adjustment against the overdue bill for which a delinquency process is created to initiate the cancellation request for an individual membership through an outbound message. In such case, the system behaves in the following manner depending on the status of the delinquency process:

- If a payment or an adjustment is made when a delinquency process is in the **Initiated, Delinquency In Progress, or On Hold** status, the status of the delinquency process is changed to the value (for example, **Canceled**) specified in the **Delinquency Process Canceled Status** attribute of the delinquency process preference. For more information on how the delinquency process is automatically canceled, refer to the [Canceling a Delinquency Process](#) section.
- If a payment or an adjustment is made when a delinquency process is in the **Pending Termination** status, the status of the delinquency process is changed to the value (for example, **Canceled**) specified in the **Delinquency Process Canceled Status** attribute of the delinquency process preference. For more information on how the delinquency process is automatically canceled, refer to the [Canceling a Delinquency Process](#) section. This scenario is addressed only when such delinquency processes are automatically canceled through the **C1-DPMON** batch.
- If a payment or an adjustment is made when a delinquency process is in the **Completed** status, no action takes place on the individual membership cancellation delinquency process.

However, if the payment or adjustment is canceled due to any reason, the system executes the **C1-CDPPAYCAN** or **C1-DPADJCAN** algorithm attached to the customer class or adjustment type, respectively, to resume the delinquency process. For more information on how the delinquency process is automatically resumed, refer to the [Resuming a Delinquency Process](#) section.

Points to Note:

If a payment or an adjustment is canceled when an individual membership cancellation delinquency process is in the **Completed** status, no action takes place on the delinquency process.

The system considers the delinquency process preference which is specified in the **Delinquency Process Field Mapping** option type of the **DELINPROC** feature configuration. For more information about the feature configuration, refer to the [DELINPROC Feature Configuration](#) section.

Individual Membership Termination Delinquency Process

There might be scenarios wherein the individual customers fail to pay premium for their health plan coverage even after sending continuous reminders. In such scenarios, you may want to terminate their individual memberships in the system. Oracle Revenue Management and Billing enables you to initiate the termination process for an individual membership through the **Delinquency Management** feature. You can configure the system such that a delinquency process is created for the individual membership's account to send a termination outbound message to the enrollment system when the overdue bills do not meet the tolerance limit. Only when an inbound message is received from the enrollment system to terminate an individual membership, the system changes the status of the individual membership to **Terminated**.

To create a delinquency process to initiate termination request for an individual membership through an outbound message, you need to design the delinquency process type (using which you want to create a delinquency process) in either of the following manner:

Delinquency Process Type	Delinquency Event Type	System Event	Algorithm	Algorithm Purpose
DP1	Send First Reminder Notification	Event Activation	C1-DLQCRTODO	Creates a To Do using the given To Do type
	Send a Warning Letter	Event Activation	C1-DLQSENDCC	Sends a warning letter via a customer contact
	Send a Termination Request	Event Activation	C1-DETERMDT	Calculates the termination date for an individual membership
		Event Activation	C1-DETRTGDT	Calculates the termination request date
		Monitor Completed Event	C1-TRNTERTRG	Updates the status of the delinquency process on the termination request date
DP2	Send First Reminder Notification	Event Activation	C1-DLQCRTODO	Creates a To Do using the given To Do type
	Send a Warning Letter	Event Activation	C1-DLQSENDCC	Sends a warning letter via a customer contact
	Calculate Termination Date	Event Activation	C1-DETERMDT	Calculates the termination date for an individual membership
		Event Activation	C1-DETRTGDT	Calculates the termination request date

Delinquency Process Type	Delinquency Event Type	System Event	Algorithm	Algorithm Purpose
	Send Termination Request	Monitor Completed Event	C1-TRNTERTRG	Updates the status of the delinquency process on the termination request date
DP3	Send First Reminder Notification	Event Activation	C1-DLQCRTODO	Creates a To Do using the given To Do type
	Send a Warning Letter	Event Activation	C1-DLQSENDCC	Sends a warning letter via a customer contact
	Calculate Termination Date	Event Activation	C1-DETTERRMDT	Calculates the termination date for an individual membership
	Send Termination Request	Event Activation	C1-DETTRTGDT	Calculates the termination request date
	Update Delinquency Process Status	Monitor Completed Event	C1-TRNTERTRG	Updates the status of the delinquency process on the termination request date

For more information about these algorithms, refer to the [Delinquency Event Type](#) section. Note that sending a To Do notification and a warning or dunning letter before initiating the termination request for an individual membership through a delinquency process is optional. You may or may not include these event types while designing a delinquency process type to initiate termination request for individual memberships billed to the delinquent account. However, you need to attach the **C1-DETTERRMDT**, **C1-DETTRTGDT**, and **C1-TRNTERTRG** algorithms to the above-mentioned system events to initiate termination request for an individual membership.

Points to Note:

A delinquency event type where the **C1-DETTRTGDT** and/or **C1-TRNTERTRG** algorithms are attached should be added at the last in the delinquency process type (as shown in the above examples – DP1, DP2, and DP3).

You can attach an algorithm to either or both of the system events (i.e. **Event Activation** and **Monitor Completed Event**) in a delinquency event type.

Once a delinquency process is created for an account, the status of the delinquency process is set to the value (for example, **Initiated**) specified in the **Delinquency Process Initiated Status** attribute of the delinquency process preference. Once the first event in the delinquency process is triggered, the status of the delinquency process is changed to the value (for example, **Delinquency In Progress**) specified in the **Delinquency Process InProgress Status** attribute of the delinquency process preference. After calculating the termination date and termination request date, the status of the respective delinquency events is changed to **Completed**. However, the status of the delinquency process changes to a status which is specified in the **Delinquency Process Pending Termination Status** attribute of the delinquency process preference only on or after the termination request date when the **C1-DPMON** batch is executed. Once the status of the delinquency process is changed to **Pending Termination**, the system executes the **C1-MEMTRMRSN** algorithm attached to the **Pending Termination** status in the lifecycle of the **C1-DelinquencyProcess** business object.

The **C1-MEMTRMRSN** algorithm updates the status reason of the active individual memberships which are billed to the delinquent account. The system considers the active individual memberships depending on the value defined for the **Skip Termination of Guaranteed Available Memberships(Valid Values-Y,N)** parameter. If the **Skip Termination of Guaranteed Available Memberships(Valid Values-Y,N)** parameter is set to **Y**, the system does not consider the active individual memberships where the **Evaluate Guaranteed Availability** flag is set to **true**. However, if the **Skip Termination of Guaranteed Available Memberships(Valid Values-Y,N)** parameter is set to **N**, the system considers the active individual memberships irrespective of whether the **Evaluate Guaranteed Availability** flag is set to **true** or **false**.

If the individual membership's start date is later than the respective termination date, this algorithm does the following:

- Updates the status reason of the individual membership. It considers the status reason which is specified in the **Awaiting Membership Cancellation Reason** attribute of the delinquency process preference. Before updating the status reason of the individual membership, the system validates whether the status reason is defined for the **Active** status of the **C1-IndMembership** business object.
- Creates a log entry for each individual membership in the delinquency process

However, if the individual membership's start date is earlier than the respective termination date and if the individual membership's end date is later than the respective termination date, this algorithm does the following:

- Updates the status reason of the individual membership. It considers the status reason which is specified in the **Membership Termination Reason for Delinquency** attribute of the delinquency process preference. Before updating the status reason of the individual membership, the system validates whether the status reason is defined for the **Active** status of the **C1-IndMembership** business object.
- Adds the termination date and termination reason (i.e. status reason) corresponding to the individual membership in the **CI_MEMBERSHIP** table.
- Sets the individual membership's end date to the termination date corresponding to the individual membership in the **CI_MEMBERSHIP** table.

- Creates a log entry in each individual membership indicating that it is terminated due a delinquency process
- Creates a log entry for each individual membership in the delinquency process
- Stamps the delinquency process ID as a characteristic using the **C1-DELPO** characteristic type on the individual membership

Once the status reason of the individual memberships is changed, you need to execute the **C1-MTOMT** batch to generate the cancellation or termination outbound messages for the individual memberships. For more information about the batch, refer to the [Outbound Message Generation for Delinquency Management](#) section. Before creating the cancellation or termination outbound messages for the individual memberships, you need to setup some pre-requisites for the cancellation or termination outbound message generation. For more information about the pre-requisites, refer to the [Membership Cancellation Outbound Message](#) and [Membership Termination Outbound Message](#) sections, respectively.

Once the cancellation or termination outbound message is sent to the enrollment system, you need to wait for the enrollment system to send the inbound message request to cancel or terminate the individual membership in ORMB. Once the inbound message to cancel or terminate an individual membership is processed, the status of the individual membership is changed to **Canceled** or **Terminated**, respectively. Once the individual membership is canceled in ORMB, the status of the delinquency process is changed to the value (for example, **Canceled**) specified in the **Delinquency Process Canceled Status** attribute of the delinquency process preference when the **C1-DPMON** batch is executed. Similarly, once the individual membership is terminated in ORMB, the status of the delinquency process is changed to the value (for example, **Customer Terminated**) specified in the **Delinquency Process Terminated Status** attribute of the delinquency process preference when the **C1-DPMON** batch is executed.

Once the status of the delinquency process is changed to **Customer Terminated**, the system executes the following algorithms (if configured) attached to the respective status of the **C1-DelinquencyProcess** business object:

- **C1-CRDPTERM** - It creates a subsequent delinquency process for the terminated customer. A subsequent delinquency process includes all the unpaid bills which were present in the primary delinquency process through which the individual membership is terminated. In ORMB, the primary delinquency process is referred to as the related delinquency process for the subsequent delinquency process. This algorithm stores the related delinquency process ID in the **REL_DELIN_PROC_ID** column corresponding to the subsequent delinquency process in the **C1_DELIN_PROC** table.
- **C1-DLQCCDPS** - It creates a customer contact to send a letter (for example, a harsh letter) to the terminated customer to communicate any penalty or legal notice on defaulting or delaying payments. If the delinquency process is created for an account, this algorithm checks whether the **Receives Notification** option is selected in the bill routing information of any person associated with the account. If the **Receives Notification** option is selected in the bill routing information of a person, this algorithm creates a customer contact for the person. Note that if the value is specified for the **Account Relationship Type** parameter, this algorithm will consider only those persons who are associated with the account using the given relationship type.

In addition, this algorithm does the following:

- Creates a log entry for each customer contact in the delinquency process.
- Stamps the delinquency process ID as a characteristic using the **C1-DELPO** characteristic type on the customer contact.
- Stamps the account ID as a characteristic using the given characteristic type on the customer contact.

The system may receive a payment or an adjustment against the overdue bill for which a delinquency process is created to initiate the termination request for an individual membership through an outbound message. In such case, the system behaves in the following manner depending on the status of the delinquency process:

- If a payment or an adjustment is made when a delinquency process is in the **Initiated, Delinquency In Progress, On Hold, or Pending Termination** status, the status of the delinquency process is changed to the value (for example, **Canceled**) specified in the **Delinquency Process Canceled Status** attribute of the delinquency process preference. For more information on how the delinquency process is automatically canceled, refer to the [Canceling a Delinquency Process](#) section.

However, if the payment or adjustment is canceled due to any reason, the system executes the **C1-CDPPAYCAN** or **C1-DPADJCAN** algorithm attached to the customer class or adjustment type, respectively, to resume the delinquency process. For more information on how the delinquency process is automatically resumed, refer to the [Resuming a Delinquency Process](#) section.

- If a payment or an adjustment is made when a delinquency process is in the **Customer Terminated** status, you need to wait till an inbound message request is received from the enrollment system to reinstate an individual membership. Once the status of the individual membership is changed to **Active**, the system does the following:
 - Changes the status of the primary delinquency process through which the termination request was initiated to a status (for example, **Customer Reinstated**) which is specified in the **Delinquency Process Customer Reinstated Status** attribute of the delinquency process preference.
 - Changes the status of the subsequent delinquency process to a status (for example, **Canceled**) which is specified in the **Delinquency Process Canceled Status** attribute of the delinquency process preference if it meets the cancel criteria

For more information on how the delinquency process is automatically canceled, refer to the [Canceling a Delinquency Process](#) section.

If the payment or adjustment is canceled due to any reason, the system executes the **C1-RODPDUEPX** or **C1-RODPONADX** algorithm attached to the customer class or adjustment type, respectively, to resume the delinquency process. For more information on how the delinquency process is automatically resumed, refer to the [Resuming a Delinquency Process](#) section.

Note: The system considers the delinquency process preference which is specified in the **Delinquency Process Field Mapping** option type of the **DELINPROC** feature configuration. For more information about the feature configuration, refer to the [DELINPROC Feature Configuration](#) section.

Advance Deposit Delinquency Process

There might be scenarios wherein the group customers want to settle their unpaid dues using the advance deposit. Oracle Revenue Management and Billing enables you to settle the unpaid bill amount of a person or an account using the advance deposit available at the parent customer or policy level through a delinquency process. Note that this feature is offered only for the fully insured group business.

To create a delinquency process through which you can apply advance deposit to settle the overdue bills of a person or an account, you need to design the delinquency process type (using which you want to create the delinquency process) in the following manner:

Delinquency Process Type	Delinquency Event Type	System Event	Algorithm	Algorithm Purpose
DP1	Apply Advance Deposit	Event Activation	C1-APPADVDEP	Transfers Advance Deposit to Pay Off Overdue Bills

The **C1-APPADVDEP** algorithm is shipped with the product. For more information about the **C1-APPADVDEP** algorithm, refer to the [Delinquency Event Type](#) section.

Once a delinquency process is created for a person or an account to apply advance deposit to settle the overdue bills, the status of the delinquency process is set to the value (for example, **Initiated**) specified in the **Delinquency Process Initiated Status** attribute of the delinquency process preference. Once the first event in the delinquency process is triggered, the status of the delinquency process is changed to the value (for example, **Delinquency In Progress**) specified in the **Delinquency Process InProgress Status** attribute of the delinquency process preference. The status remains as **Delinquency In Progress** even after the offset request adjustments are created through the first or subsequent event in the delinquency process. On executing the **C1-DPMON** batch, the system checks whether the delinquency process meets the cancel criteria. If so, the system will change the status of the delinquency process to a status (for example, **Canceled**) which is specified in the **Delinquency Process Canceled Status** attribute of the delinquency process preference.

The system may receive a payment or an adjustment against the overdue bill for which an advance deposit delinquency process is created. In such case, the system behaves in the following manner depending on the status of the delinquency process:

- If a payment or an adjustment is made when a delinquency process is in the **Initiated**, **Delinquency In Progress**, or **On Hold** status, the status of debit adjustments (if any) made against the payments, the status of the credit adjustment (if any) used to offset the overdue bills, and the status of the offset request adjustments (if any) is changed to **Canceled**. The status of the delinquency process is changed to the value (for example, **Canceled**) specified in the **Delinquency Process Canceled Status** attribute of the delinquency process preference. In addition, the status of the offset request (if any) is changed to **Unapplied Offset** or **Defer Unapplied**. For more information, refer to the [Canceling a Delinquency Process](#) and [Notification on Advance Deposit Cancellation](#) sections.
- If a payment or an adjustment is made when an advance deposit delinquency process is in the **Canceled** status, no action takes place on the advance deposit delinquency process.

However, if the payment or adjustment is canceled due to any reason, the system executes the **C1-CDPPAYCAN** or **C1-DPADJCAN** algorithm attached to the customer class or adjustment type, respectively, to resume the delinquency process. For more information on how the delinquency process is automatically resumed, refer to the [Resuming a Delinquency Process](#) section.

Points to Note:

If a payment or an adjustment is canceled when an advance deposit delinquency process is in the **Canceled** status, no action takes place on the advance deposit delinquency process.

The system considers the delinquency process preference which is specified in the **Delinquency Process Field Mapping** option type of the **DELINPROC** feature configuration. For more information about the feature configuration, refer to the [DELINPROC Feature Configuration](#) section.

Delinquency Control

You need to define a delinquency control for each collection class where the collection method is set to **Self-Control Delinquency**. A new maintenance object named **C1-DELIN-CTR**, a new business object named **C1-DelinControl**, and a new screen named **Delinquency Control** are introduced in this release.

The **Delinquency Control** screen enables you to create, edit, and delete a delinquency control. Note that this screen enables you to define a delinquency control only for those collection classes where the collection method is set to **Self-Control Delinquency**.

A delinquency control enables the system to determine the following:

- A minimum credit review frequency at which the persons or accounts belonging to the respective collection class should be monitored for delinquency. In other words, it indicates in how many days the system should review whether the bills of the persons or accounts are overdue.
- A delinquency monitor rule algorithm that you want to invoke for the collection class while executing the **C1-PRDLQ** or **C1-ACDLQ** batch.

A new algorithm type named **C1-DLNQMNRL** is introduced in this release. This algorithm creates a new delinquency process for a person or an account when the person's or account's debt is overdue and does not meet the tolerance limit. Before creating a new delinquency process for a person or an account, it checks whether a delinquency process in the given status already exists for the person or the account. If a delinquency process already exists for the person or the account in the given status, the system will add new overdue bills in the existing delinquency process and will not create a new delinquency process for the person or the account. While adding a bill to an existing delinquency process, the system enables you to create a To Do for the delinquency process.

While executing the **C1-PRDLQ** or **C1-ACDLQ** batch, the system derives the delinquency monitor rule algorithm attached to the delinquency control of the respective collection class for each derived person or account, respectively. This algorithm then determines whether the person's or account's debt should be monitored for delinquency based on their respective credit rating. The system considers the sum of account's credit rating and credit rating threshold (defined in the installation options) to determine whether the account's debt should be monitored for delinquency. However, the system considers the sum of person's credit rating and credit rating threshold (defined in the installation options) to determine whether determine whether the person's debt should be monitored for delinquency.

If you specify the value for the **Only Process Customers with Credit Rating Less than or Equal to This Value** parameter, the system checks whether person's or account's credit rating is less than or equal to the specified value. If the person's or account's credit rating is greater than the specified value, the system does not monitor the person's or account's debt for delinquency. However, if the person's or account's credit rating is less than or equal to the specified value or if the value is not specified for the **Only Process Customers with Credit Rating Less than or Equal to This Value** parameter, the system monitors the person's or account's debt for delinquency.

While monitoring the account's debt for delinquency, this algorithm fetches all the completed bills of the account which are unpaid and not yet linked to any delinquency process of the account. Here, the system considers the delinquency process of the account which is in the non-final status. While monitoring the person's debt for delinquency, this algorithm fetches all the unpaid bills of the person's accounts and all the unpaid bills of the child persons' accounts. It considers only those accounts of the person, its child persons, and the child persons' accounts where the collection method in the respective collection class is set to **Parental Delinquency** and not where the collection method in the respective collection class is set to **Self-Control Delinquency** or **Not Eligible for Collection**. In addition, it considers only those child persons who are associated with the parent person using a person relationship type which is not specified in the **Person Relationship Type Exclusion List** parameter. Note that if an active promise to pay exists for an account, the system does not monitor the delinquency of the account.

While fetching the unpaid bills of the accounts, the system checks the following:

- Whether the bill is included in any payment agreement request which is in the non-final status. If the bill is included in any payment agreement request which is in the non-final status, the system does not consider the overdue bill for monitoring the person or account's debt. However, if the bill is not included in any payment agreement request which is in the non-final status, the system considers the overdue bill for monitoring the person or account's debt.
- Whether a characteristic using the characteristic type specified in the **Postpone Date Characteristic Type** parameter is defined for a bill. If the postpone overdue review until date is specified for the bill, the system checks whether postpone overdue review date of the bill is later than the batch business date. If the postpone overdue review date of the bill is later than the batch business date, the system does not consider the overdue bill for monitoring the person or account's debt. However, if the postpone overdue review until date is not specified for the bill or if the postpone overdue review date of the bill is earlier than or equal to the batch business date, the system considers the overdue bill for monitoring the person or account's debt.

- Whether the age of a bill is greater than or equal to the value specified in the **Only Process Bills with an Age Greater than or Equal to This value** parameter. The system calculates the age of a bill using the value specified in the **Which Date Determines Age ('D' – Due Date or 'B' – Bill Date)** parameter. If the parameter value is set to **B**, the system calculates the invoice age from the bill date until the system date. However, if the parameter value is set to **D**, the system calculates the invoice age from the bill due date until the system date. If the invoice age is greater than or equal to the value specified in the **Only Process Bills with an Age Greater than or Equal to This value** parameter, the system considers the overdue bill for monitoring the person or account's debt. However, if the invoice age is less than the value specified in the **Only Process Bills with an Age Greater than or Equal to This value** parameter, the system does not consider the overdue bill for monitoring the person or account's debt.

Note that the system validates the former two conditions on both the debit and credit bills of the account. However, the system validates the age of a debit bill and not of a credit bill.

Once the list of overdue bills is finalized, the system derives the original and unpaid amounts for each overdue bill of the person or account whose debt is monitored. While monitoring the debt of an account, the system calculates the following:

- Sum of the unpaid amounts of the account's bills (that meet the above criteria)
- Sum of the original amounts of the account's bills (that meet the above criteria)

However, while monitoring the debt of a person, the system calculates the following:

- Sum of the unpaid amounts of the bills (that meet the above criteria) of the person's derived accounts (including the ones from its hierarchy)
- Sum of the original amounts of the bills (that meet the above criteria) of the person's derived accounts (including the ones from its hierarchy)

While monitoring the debt of a person, the system also checks whether the **Include On Account Payments In Threshold Evaluation - Y/N** parameter is set to **Y** or **N**. If the **Include On Account Payments In Threshold Evaluation - Y/N** parameter is set to **Y**, the system derives all the on account payments which are made against the on account payment contracts. The system considers the on account payment contracts which are created using the contract types that are specified in the **On Account Payment Contract Types** parameter. Once the on account payments (if any) are derived for the person, the system deducts the on account payments of the customer from the total unpaid amount of the customer.

Note that, while calculating the sum of the original amounts, the system considers the **Use Current Revenue Period Billed For Latest Overdue Due Date – Y/N** parameter. If the **Use Current Revenue Period Billed For Latest Overdue Due Date – Y/N** parameter is set to **Y**, the system considers all those financial transactions of the account whose coverage start date is earlier than or equal to the bill due date while calculating the total billed (i.e. original) amount. This option helps to handle the scenario wherein an individual short pays by just enough each month but still does not become delinquent. This is because, in this case, the system calculates the unpaid percentage or amount based on the total amount billed for the coverage period rather than the total amount billed for the overdue bills.

Once the total unpaid amount and the total billed amount is calculated for the person or account, the system checks the value of the **Unpaid Amount and Percentage Required ('Y' or 'N')** parameter. If the **Unpaid Amount and Percentage Required ('Y' or 'N')** parameter is set to **Y** or **N**, the system checks the following:

- Total unpaid amount is greater than or equal to the tolerance limit
- Unpaid percentage value (i.e. unpaid amount with respect to original amount) is greater than or equal to the tolerance limit

If the **Unpaid Amount and Percentage Required ('Y' or 'N')** parameter is set to **Y**, the person or account should meet both the above two conditions. The system then considers the person or account as delinquent. However, if the **Unpaid Amount and Percentage Required ('Y' or 'N')** parameter is set to **N**, the person or account should meet either of the above two conditions. The system then considers the person or account as delinquent.

Once the person or account is considered as delinquent, this algorithm creates a new delinquency process using the delinquency process type which is specified in the given delinquency process type preference. You can specify two separate delinquency process type preference – one for the fully insured group business and another for the fully insured individual business. Depending on whether the person or account is linked to policies or individual memberships, the system will accordingly refer the given delinquency process type preference. For more information about the delinquency process type preference, refer to the [Delinquency Process Type Preference](#) section. The status of the new delinquency process is set to the value (for example, **Initiated**) specified in the **Delinquency Process Initiated Status** attribute of the delinquency process preference. Note that if the delinquency process already exists for the person or account in the given status, then the system will add new overdue bills in the existing delinquency process and will not create a new delinquency process for the person or account. While adding a bill to an existing delinquency process, the system enables you to create a To Do for the delinquency process.

Note: The system considers the delinquency process preference which is specified in the **Delinquency Process Field Mapping** option type of the **DELINPROC** feature configuration. For more information about the feature configuration, refer to the [DELINPROC Feature Configuration](#) section.

In addition, this algorithm adds a log entry for each overdue bill in the delinquency process. This algorithm contains the following parameters:

- **Only Process Customers with Credit Rating Less than or Equal to This Value** – Used when you want to initiate the delinquency process for only those persons or accounts whose credit rating is less than or equal to the specified value. This parameter is optional.
- **Only Process Bills with an Age Greater than or Equal to This value** – Used to specify the number of days. The system considers a bill while monitoring the person or account's debt when its age is greater than or equal to the specified number of days. This parameter is optional.
- **Which Date Determines Age ('D' - Due Date or 'B' - Bill Date)** – Used to indicate whether you want the system to determine the age of the invoice (i.e. bill) using the due date or the bill date. The valid values are **D** and **B**. This parameter is mandatory.

- **Use Current Revenue Period Billed For Latest Overdue Due Date - Y/N** – Used when you want to consider all those financial transactions of the account whose coverage start date is earlier than or equal to the bill due date while calculating the total billed (i.e. original) amount. The valid values are **Y** and **N**. This parameter is optional. If you do not specify the parameter value, by default, it is set to **N**.
- **Include On Account Payments In Threshold Evaluation - Y/N** – Used to indicate whether you want to consider the payments made against the on account contracts of the account while calculating the unpaid amount. The valid values are **Y** and **N**. This parameter is optional. If you do not specify the parameter value, by default, it is set to **N**.
- **On Account Payment Contract Types** – Used to indicate the on account contract types that you want to consider while determining the total unpaid amount. Here, you must specify a contract type which is already defined in the system. You can specify maximum twenty comma-separated values for this parameter. This parameter is only required when the **Include On Account Payments In Threshold Evaluation - Y/N** parameter is set to **Y**.
- **Only Process Person/Account with an Unpaid Percentage Greater Than or Equal to This Value** – Used to specify the unpaid percentage value (i.e. unpaid amount with respect to original amount). The system then considers a person or an account as delinquent only when the unpaid percentage of its overdue bills is greater than or equal to the specified value. This parameter is required.
- **Only Process Person/Account with an Unpaid Amount Greater Than or Equal to This Value** - Used to specify the flat unpaid amount. The system then considers a person or an account as delinquent only when the unpaid amount of its overdue bills is greater than or equal to the specified value. This parameter is required.
- **Person Relationship Type Exclusion List** – Used when you do not want to consider the child persons who are associated with the parent person using a particular person relationship type while fetching the unpaid bills from the person's hierarchy. Here, you must specify a person relationship type which is already defined in the system. You can specify maximum 10 comma-separated values for this parameter. This parameter is only applicable for the fully insured group business and not for the fully insured individual business.
- **To Do Entry Required For Delinquency** – Used to indicate whether you want the system to create a To Do when a new overdue bill is added to an existing delinquency process. The valid values are **Y** and **N**. This parameter is mandatory.
- **To Do Type of To Do Entry for Delinquency** – Used to indicate the To Do type using which you want to create the To Do for the existing delinquency process. Here, you must specify a To Do type which is already defined in the system. This parameter is only required when the **To Do Entry Required For Delinquency** parameter is set to **Y**.
- **Delinquency Process Status For Existing Delinquency Process** – Used to specify the status of the delinquency process. The system then checks whether any delinquency process exists for the person or account in the specified status. If so, the system adds a new overdue bill in the respective delinquency process and does not create a new delinquency process for the person or account. You can specify comma-separated values for this parameter. This parameter is mandatory.

Note: The system allows you to add new bills to a delinquency process when it is in any of the following three statuses – **INITIATED** (i.e. **Initiated**), **INPROGRESS** (i.e. **Delinquency In Progress**), and **HOLD (On Hold)**. Therefore, you need to accordingly specify the value in the **Delinquency Process Status For Existing Delinquency Process** parameter.

- **Postpone Date Characteristic Type** – Used to indicate a characteristic type which is used to store the postpone overdue review until date for a bill. Here, you must specify a characteristic type where the characteristic entity is set to **Bill**. The **ADMRVWDT** characteristic type is shipped with the product. This parameter is optional.

Note: You have an option to set the specified characteristic type on the bill if you want to postpone the overdue review date of the bill. If the specified characteristic type is defined on the unpaid bill and if the postpone overdue review until date is later than the batch business date, the system does not consider the overdue bill for monitoring the person or account's debt.

- **Unpaid Amount and Percentage Required ('Y' or 'N')** – Used to indicate whether you want to consider both the unpaid amount and the unpaid percentage to check whether the person or account is delinquent. The valid values are:
 - **Y** - If you set the value to **Y**, the system considers both the unpaid amount and the unpaid percentage to check whether the person or account is delinquent.
 - **N** - If you set the value to **N**, the system considers either the unpaid amount or the unpaid percentage to check whether the person or account is delinquent.

This parameter is mandatory.

- **Delinquency Process Type Field Mapping – Individual** – Used to specify a preference where the preference category is set to **Delinquency Process Type**. The system uses the preference to determine the delinquency process type using which the delinquency process should be created for an account to which individual memberships are billed. This parameter is mandatory. It is not applicable for the fully insured group business. For more information about the delinquency process type preference, refer to the [Delinquency Process Type Preference](#) section.
- **Delinquency Process Type Field Mapping – Group** – Used to specify a preference where the preference category is set to **Delinquency Process Type**. The system uses the preference to determine the delinquency process type using which the delinquency process should be created for a person or an account to which policies are billed. This parameter is mandatory. It is not applicable for the fully insured individual business. For more information about the delinquency process type preference, refer to the [Delinquency Process Type Preference](#) section.

Delinquency Event Type

Once you create a delinquency control for each collection class where the collection method is set to **Self-Control Delinquency**, you need to create the required delinquency event types using which you want the system to create delinquency events in the delinquency process. If an account is delinquent, you may want to send different kinds of letters such as dunning letter, warning letter, etc., send a To Do notification, initiate the termination or cancellation request for the individual memberships which are billed to the delinquent account, or initiate the termination or reinstatement request for the policies which are billed to the delinquent account. However, if a person is delinquent, you may want to send different kinds of letters such as dunning letter, warning letter, etc., send a To Do notification, or initiate the termination or reinstatement request for the policies which are billed to the delinquent person. The system enables you to send different kinds of letters, send To Do notification, or initiate the termination, cancellation, or reinstatement request by triggering different delinquency events at predefined intervals in the delinquency process.

A new maintenance object named **C1-DLQ-EV-TP**, a new business object named **C1-DelinEventType**, and a new screen named **Delinquency Event Type** are introduced in this release. The **Delinquency Event Type** screen enables you to create, edit, delete, and copy a delinquency event type. A delinquency event type enables the system to determine the following:

- The primary actions that you want to perform when the delinquency event of the respective delinquency event type is triggered manually (on or before the trigger date from the user interface) or automatically (on the trigger date through the **C1-DPMON** batch). For example, a primary action can be sending a letter, sending a To Do notification, initiating a cancellation, termination, or reinstatement request, etc. You can configure the system to perform the primary actions on the **Event Activation** system event.
- The secondary actions are those actions that you want to perform when the primary actions are completed. For example, a secondary action can be changing the delinquency process status, recalculating the trigger dates of the remaining events in the delinquency process, etc. You can configure the system to perform the secondary actions on the **Monitor Completed Event** system event.

Note that you can attach an algorithm to either or both of the system events while defining a delinquency event type. Once the algorithms attached to the **Event Activation** system event are executed, the status of the delinquency event is changed to **Completed**. The algorithms attached to the **Monitor Completed Event** system event are not executed in the online mode when the status of the current or prior delinquency event is changed to **Completed**. These algorithms are triggered when you execute the **C1-DPMON** batch.

Whenever you manually execute the subsequent events (non-first event) in the delinquency process, the system first checks whether the algorithms attached to the **Monitor Completed Event** system event of the previous delinquency event is successfully executed or not. If an algorithm attached to the **Monitor Completed Event** system event of the previous delinquency event is not executed successfully, the system displays a warning message that indicates the previous event's monitoring algorithms are not executed successfully and confirms whether you want to continue in such scenario. If you click **Yes**, the system ignores the previous event's monitoring algorithm and executes the algorithms attached to the **Event Activation** system event of the current delinquency event (which is triggered). However, if you click **No**, the system does not execute the algorithms attached to the **Event Activation** system event of the current delinquency event. Note that if there are no algorithms attached to the **Monitor Completed Event** system event of the previous delinquency event, the system will directly execute the algorithms attached to the **Event Activation** system event of the current delinquency event.

Delinquency Event Activation

The following table lists the primary actions that you can perform using the respective algorithm shipped with the product:

Action	Algorithm	Algorithm Description
Create a To Do Notification	C1-DLQCRTODO	<p>This algorithm creates a To Do for the following:</p> <ul style="list-style-type: none"> • Person (i.e. parent customer or bill group) when the delinquency process is created for a person • Main customer of the account when the delinquency process is created for an account <p>It also creates a log entry for each To Do notification in the delinquency process. This algorithm is invoked when the event is triggered manually from the user interface or automatically through the C1-DPMON batch.</p> <p>You can use this algorithm in a delinquency process type using which you want to create a delinquency process through which To Do notifications are sent to the delinquent customer or account. It should be attached to the delinquency event type which sends a To Do notification to the person or the main customer of the account.</p> <div style="border: 1px solid black; padding: 5px;"> <p>Note: Once a To Do notification is created, a record is added in the C1_DELIN_PROC_EVT_NOTIF table where the DP_EVT_NOTIF_TYPE_FLG column is set to TODO and the DP_EVT_NOTIF_ID column is set to the To Do Entry ID.</p> </div> <p>This algorithm contains the following parameters:</p> <ul style="list-style-type: none"> • To Do Role Person – Used to indicate that users with the specified To Do role must receive the To Do notification when the delinquency process is created for a person. This parameter is optional. If you do not specify this parameter, the system considers the default To Do role associated with the To Do type. This parameter is only applicable for the fully insured group business. • To Do Type Person – Used to indicate the To Do type using which you want to create a To Do when the delinquency process is created for a person. The C1-DLQEV To Do type is shipped with the product to create To Dos for the person level delinquency process.

Action	Algorithm	Algorithm Description
		<p>This parameter is mandatory. This parameter is only applicable for the fully insured group business.</p> <ul style="list-style-type: none"> • To Do Role Account – Used to indicate that users with the specified To Do role must receive the To Do notification when the delinquency process is created for an account. This parameter is optional. If you do not specify this parameter, the system considers the default To Do role associated with the To Do type. This parameter is applicable for both the fully insured group and fully insured individual businesses. • To Do Type Account – Used to indicate the To Do type using which you want to create a To Do when the delinquency process is created for an account. The C1-DLQAC To Do type is shipped with the product to create To Dos for the account level delinquency process. This parameter is mandatory. This parameter is applicable for both the fully insured group and fully insured individual businesses.
Send Letter via Customer Contact	C1-DLQSENDCC	<p>This algorithm creates a customer contact to send a letter (for example, dunning letter, warning letter, etc.). If the delinquency process is created for a person, this algorithm checks whether the Notify Group Customer Or Bill Groups Or Bill Accounts parameter is set to PG, BG, or BA. Depending on the value defined for the Notify Group Customer Or Bill Groups Or Bill Accounts parameter, this algorithm does the following:</p> <ul style="list-style-type: none"> • If the Notify Group Customer Or Bill Groups Or Bill Accounts parameter is set to PG, this algorithm checks whether the person type of the person is set to Parent Customer or Bill Group. If the person type of the person is set to Parent Customer, this algorithm creates a customer contact for the parent customer. However, if the person type of the person is set to Bill Group, this algorithm derives the parent customer of the bill group (using the Person Relationship Type option type of the C1-ASOBLNG feature configuration) and then creates a customer contact for the parent customer.

Action	Algorithm	Algorithm Description
		<ul style="list-style-type: none"> • If the Notify Group Customer Or Bill Groups Or Bill Accounts parameter is set to BG, this algorithm checks whether the person type of the person is set to Parent Customer or Bill Group. If the person type of the person is set to Parent Customer, this algorithm creates a customer contact for the parent customer and for all those bill groups of the parent customer who are associated with the parent customer using the relationship type which is specified in the Person Relationship Type option type of the C1-ASOBLNG feature configuration. However, if the person type of the person is set to Bill Group, this algorithm derives the parent customer of the bill group (using the Person Relationship Type option type of the C1-ASOBLNG feature configuration) and then creates a customer contact for the parent customer and for the respective bill group. • If the Notify Group Customer Or Bill Groups Or Bill Accounts parameter is set to BA, this algorithm checks whether the person type of the person is set to Parent Customer or Bill Group. If the person type of the person is set to Parent Customer, this algorithm creates a customer contact for each account to which the parent customer and its bill groups are associated as the main customer. The system considers those bill groups of the parent customer who are associated with the parent customer using the relationship type which is specified in the Person Relationship Type option type of the C1-ASOBLNG feature configuration. However, if the person type of the person is set to Bill Group, this algorithm derives the parent customer of the bill group (using the Person Relationship Type option type of the C1-ASOBLNG feature configuration) and then creates a customer contact for each account to which the parent customer and the respective bill group are associated as the main customer. However, if the delinquency process is created for an account, this algorithm checks whether the Receives Notification option is selected in the bill routing information of any person associated with the account.

Action	Algorithm	Algorithm Description
		<p>If the Receives Notification option is selected in the bill routing information of a person, this algorithm creates a customer contact for the person. Note that if the value is specified for the Account Relationship Type parameter, this algorithm will consider only those persons who are associated with the account using the given relationship type. Irrespective of whether the delinquency process is created for a person or an account, this algorithm also checks whether the Send Membership Level Notification Source field corresponding to the respective delinquency event type in the respective delinquency process type is set to Delinquency Process Type or Algorithm. If the Send Membership Level Notification Source field is set to Delinquency Process Type, this algorithm checks whether the Send Membership Level Notification field corresponding to the respective delinquency event type in the respective delinquency process type is set to Send Membership Level Notification Required or Send Membership Level Notification Not Required.</p> <p>However, if the Send Membership Level Notification Source field is set to Algorithm, this algorithm invokes the C1-DERMLNBR algorithm attached to the Send Member Level Notification Option Derivation system event in the respective delinquency process type. For more information about the C1-DERMLNBR algorithm, refer to the Delinquency Process Type Algorithms section. If the value of the Send Membership Level Notification field is set to Send Membership Level Notification Required either in the delinquency process type or in the business rule depending on the send membership level notification source, this algorithm derives the memberships billed to the delinquent customer or account. It then checks whether the number of memberships derived is less than or equal to the number specified in the Member Level Notification Deferred Processing Threshold parameter. If the number of memberships derived is less than or equal to the specified value, this algorithm creates a customer contact for the main subscriber of each derived membership in the real time (i.e. immediately). In addition, the status of the delinquency event is changed to Complete.</p>

Action	Algorithm	Algorithm Description
		<p>However, if no value is defined for the Member Level Notification Deferred Processing Threshold parameter or if the number of memberships derived is greater than the specified value, this algorithm changes the status of the delinquency event to Pending Contact Creation. For more information on how the customer contacts are created at the membership level for such events, refer to the Creating Customer Contacts for Group or Individual Memberships section.</p> <p>In addition, this algorithm does the following:</p> <ul style="list-style-type: none"> • Creates a log entry for each customer contact in the delinquency process. • Stamps the delinquency process ID as a characteristic using the C1-DELPO characteristic type on the customer contact. • Stamps the account ID as a characteristic using the given characteristic type on the customer contact. <div data-bbox="748 905 1422 1220" style="border: 1px solid black; padding: 5px;"> <p>Note: The account ID is stamped as a characteristic on the customer contact only when either of the following condition is met:</p> <p>>> The delinquency process is created for a person and the Notify Group Customer Or Bill Groups Or Bill Accounts parameter is set to BA.</p> <p>>> The delinquency process is created for an account.</p> </div> <ul style="list-style-type: none"> • Stamps the membership ID as a characteristic using the given characteristic type on the customer contact. <div data-bbox="748 1367 1422 1524" style="border: 1px solid black; padding: 5px;"> <p>Note: The membership ID is stamped as a characteristic on the customer contact only when the customer contact is created for the main subscriber of the membership.</p> </div> <p>This algorithm is invoked when the event is triggered manually from the user interface or automatically through the C1-DPMON batch. You can use this algorithm in a delinquency process type using which you want to create a delinquency process through which debt collection letters are sent to the delinquent customer or account. It should be attached to the delinquency event type which sends a warning or dunning letter.</p>

Action	Algorithm	Algorithm Description
		<div data-bbox="753 262 1401 457" style="border: 1px solid black; padding: 5px;"> <p>Note: Once a customer contact is created, a record is added in the C1_DELIN_PROC_EVT_NOTIF table where the DP_EVT_NOTIF_TYPE_FLG column is set to CC and the DP_EVT_NOTIF_ID column is set to the customer contact ID.</p> </div> <p>This algorithm derives the preferred contact method for the customer contact. While deriving the preferred contact method, the algorithm first fetches the main customer’s bill route type on the account and then fetches the bill routing method of the bill route type. Once the bill routing method is derived, the system considers the preference which is specified in the Routing Method – Contact Method Mapping option type of the DELINPROC feature configuration. For more information about the feature configuration, refer to the DELINPROC Feature Configuration section.</p> <p>It then checks whether any preferred contact method is mapped to the bill routing method in the preference. If the preferred contact method is mapped to the bill routing method in the preference, the system stamps it on the customer contact. However, if the preferred contact method is not mapped to the bill routing method in the preference, the system considers the default contact method as the preferred contact method and stamps it on the customer contact.</p> <p>This algorithm contains the following parameters:</p> <ul style="list-style-type: none"> • Customer Contact Type – Used to indicate the customer contact type using which you want to create the customer contact. This parameter is mandatory. Here, you must specify a customer contact type with the appropriate letter template depending on the type of letter you want to send to the customer. • Customer Contact Class – Used to indicate the customer contact class to which the customer contact type belongs. This parameter is mandatory. • Default Contact Method – Used to indicate the contact method that you want to use when the preferred contact method cannot be derived for the routing method from the preference. This parameter is mandatory.

Action	Algorithm	Algorithm Description
		<ul style="list-style-type: none"> • Notify Group Customer Or Bill Groups Or Bill Accounts – Used to indicate whether a customer contact should be created for a parent customer, bill group, and/or for each billing account (that meets the criteria). The valid values are: <ul style="list-style-type: none"> ○ PG – Used when you want to create a customer contact only for the parent customer. ○ BG – Used when you want to create a customer contact for the parent customer and for the bill groups of the parent customer. If the person type of the person is set to Parent Customer, the system creates a customer contact for the parent customer and for all those bill groups that are associated with the parent customer using the relationship type specified in the Person Relationship Type option type of the C1-ASOBLNG feature configuration. However, if the person type of the person is set to Bill Group, the system creates a customer contact for the parent customer and the respective bill group. ○ BA – Used when you want to create a customer contact for the main customer of the account. If the person type of the person is set to Parent Customer, the system creates a customer contact for each account to which the parent customer and its bill groups are associated as the main customer. However, if the person type of the person is set to Bill Group, the system creates a customer contact for each account to which the parent customer and the respective bill group are associated as the main customer. <p>Note that this parameter is only applicable for the fully insured group business.</p>

Action	Algorithm	Algorithm Description
		<ul style="list-style-type: none"> • Account Characteristic Type – Used to indicate the characteristic type using which you want to store the account ID on the customer contact. Here, you must specify a characteristic type where the characteristic entity is set to Customer Contact. • Membership Characteristic Type – Used to indicate the characteristic type using which you want to store the membership ID on the customer contact. Here, you must specify a characteristic type where the characteristic entity is set to Customer Contact. • Account Relationship Type – Used to indicate the relationship type using which the person is associated with the account. The system will then create the customer contacts only for those persons who are associated with the account using the specified relationship type. This parameter is optional. You can specify maximum ten comma-separated values for this parameter. If you do not specify the parameter value, the system will create customer contacts for all the persons who are associated with the account irrespective of their relationship type. Note that this parameter is only applicable for the fully insured individual business. • Member Level Notification Deferred Processing Threshold – Used to specify the maximum number of memberships for which customer contacts can be created in the real time (i.e. immediately). If the number of memberships billed to the delinquent customer or account exceeds the specified limit, a customer contact is created for the main subscriber of each derived membership in the deferred mode (i.e. through the C1-MBRNT batch control).
Calculate Termination Date	C1-DETERMMDT	<p>This algorithm is used to calculate the termination date for the policies or individual memberships. This date is sent to the enrollment system through the termination outbound message. This algorithm is invoked when the event is triggered manually from the user interface or automatically through the C1-DPMON batch.</p> <p>You must use this algorithm in a delinquency process type using which you want to create a delinquency process to initiate termination request for a policy or individual membership through an outbound message.</p>

Action	Algorithm	Algorithm Description
		<p>It should be attached to the delinquency event type which requests to terminate the following:</p> <ul style="list-style-type: none"> • Policies billed to the delinquent person or account • Individual memberships billed to the delinquent account <p>It calculates the termination date based on the termination date rule. The system derives the termination date rule using an algorithm which is attached to the Termination Date Rule Derivation system event of the delinquency process type.</p> <p>In addition, this algorithm retrieves the business rules of the Delinquency Miscellaneous Options Business Rule category which are effective on the trigger date. Note that the system considers only those effective business rules which are in the Active status. Once the effective business rules are identified, the system executes these business rules in the order of their priority.</p> <p>Once a policy or individual membership meets the criteria defined in the business rule, the system refers the value of either the following parameter depending on the termination date rule (as listed in the below table) for the policy or individual membership from the business rule:</p> <ul style="list-style-type: none"> • Add Days to Coverage End Date • Add Days to Paid Through Date • Add Months to Paid Through Date • Add Days to Grace End Date • Add Days to Date of Processing of Termination Request <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>Points to Note:</p> <p>The system derives the policies using an algorithm which is attached to the Policies Derivation system event of the delinquency process type.</p> <p>The system derives the individual memberships using an algorithm which is attached to the Memberships Derivation system event of the delinquency process type.</p> </div>

Action	Algorithm	Algorithm Description												
		<p>While deriving an active individual membership, the system considers the status code which is specified in the Membership Active Status option type of the C1-ASOBLNG feature configuration. Similarly, while deriving an active policy, the system considers the status code which is specified in the Policy Active Status option type of the C1-ASOBLNG feature configuration.</p> <p>Once the system derives the termination date rule and the required parameter from the delinquency miscellaneous options business rule, the system calculates the termination date for each policy or individual membership in the following manner:</p> <table border="1" data-bbox="753 800 1412 1890"> <thead> <tr> <th data-bbox="758 806 959 957">Termination Date Rule is set to...</th> <th data-bbox="964 806 1182 957">Parameter which is used from the business rule...</th> <th data-bbox="1187 806 1404 957">Termination date is set to...</th> </tr> </thead> <tbody> <tr> <td data-bbox="758 963 959 1360">Latest Billed Coverage End Date</td> <td data-bbox="964 963 1182 1360">Add Days to Coverage End Date</td> <td data-bbox="1187 963 1404 1360">Latest Billed Coverage End Date (i.e. coverage end date that falls on or prior to the grace end date) + Add Days to Coverage End Date</td> </tr> <tr> <td data-bbox="758 1367 959 1730">Latest Due Coverage End Date</td> <td data-bbox="964 1367 1182 1730">Add Days to Coverage End Date</td> <td data-bbox="1187 1367 1404 1730">Latest Due Coverage End Date (i.e. coverage end date that falls prior to the latest due date) + Add Days to Coverage End Date</td> </tr> <tr> <td data-bbox="758 1736 959 1887">X days After Paid Through Date</td> <td data-bbox="964 1736 1182 1887">Add Days to Paid Through Date</td> <td data-bbox="1187 1736 1404 1887">Paid Through Date + Add Days to Paid Through Date</td> </tr> </tbody> </table>	Termination Date Rule is set to...	Parameter which is used from the business rule...	Termination date is set to...	Latest Billed Coverage End Date	Add Days to Coverage End Date	Latest Billed Coverage End Date (i.e. coverage end date that falls on or prior to the grace end date) + Add Days to Coverage End Date	Latest Due Coverage End Date	Add Days to Coverage End Date	Latest Due Coverage End Date (i.e. coverage end date that falls prior to the latest due date) + Add Days to Coverage End Date	X days After Paid Through Date	Add Days to Paid Through Date	Paid Through Date + Add Days to Paid Through Date
Termination Date Rule is set to...	Parameter which is used from the business rule...	Termination date is set to...												
Latest Billed Coverage End Date	Add Days to Coverage End Date	Latest Billed Coverage End Date (i.e. coverage end date that falls on or prior to the grace end date) + Add Days to Coverage End Date												
Latest Due Coverage End Date	Add Days to Coverage End Date	Latest Due Coverage End Date (i.e. coverage end date that falls prior to the latest due date) + Add Days to Coverage End Date												
X days After Paid Through Date	Add Days to Paid Through Date	Paid Through Date + Add Days to Paid Through Date												

Action	Algorithm	Algorithm Description		
		X Months After Paid Through Date	Add Months to Paid Through Date	Paid Through Date + Add Months to Paid Through Date
		Month End of Grace Period Start	-	Last Day of Grace Period Start Month
		Month End of Termination Letter Creation	-	Last Day of Termination Letter Creation Event Trigger Month
		Month End of Termination Request Event	-	Last Day of Send Termination Request Event Trigger Month
		X days After Grace End Date	Add Days to Grace End Date	Grace End Date + Add Days to Grace End Date
		X days After Termination Request Event	Add Days to Date of Processing of Termination Request	Send Termination Request Event Trigger Date + Add Days to Date of Processing of Termination Request
		<p>Note: The system retrieves the grace start date and grace end date of the delinquency process from the C1_DELIN_PROC table.</p>		
		<p>Once the termination date is calculated for each policy or individual membership, the system checks whether the termination date is earlier than the respective policy or individual membership start date. If the termination date is earlier than the respective policy or individual membership start date, the system sets the termination date to the respective policy or individual membership start date.</p>		

Action	Algorithm	Algorithm Description
		<p>In addition, if the delinquency category of the delinquency process is set to GRUP, this algorithm derives the status reason from the Policy Termination Reason for Delinquency Process attribute of the delinquency process preference. It validates whether the status reason is defined for the Terminated status of the C1-POLICY business object. Similarly, if the delinquency category of the delinquency process is set to INDV, this algorithm derives the status reason from the Membership Termination Reason for Delinquency attribute of the delinquency process preference. It validates whether the status reason is defined for the Active status of the C1-IndMembership business object.</p> <div style="border: 1px solid black; padding: 5px;"> <p>Note: The system considers the delinquency process preference which is specified in the Delinquency Process Field Mapping option type of the DELINPROC feature configuration. For more information about the feature configuration, refer to the DELINPROC Feature Configuration section.</p> </div> <p>Finally, it stores the terminate date rule, termination date, and the termination reason of each delinquency process corresponding to the record in the C1_DELIN_PROC table.</p>
Send Termination Request	C1-DETRTGDT	<p>This algorithm is used to calculate the termination request date for the delinquency event. The termination request date is the date when the termination request is initiated for the policies or individual memberships in the system and thereby the termination outbound message is sent to the enrollment system. This algorithm is invoked when the event is triggered manually from the user interface or automatically through the C1-DPMON batch.</p> <p>You must use this algorithm in a delinquency process type using which you want to create a delinquency process to initiate termination request for a policy or individual membership through an outbound message. It should be attached to the delinquency event type which requests to terminate the following:</p> <ul style="list-style-type: none"> • Policies billed to the delinquent person or account • Individual memberships billed to the delinquent account

Action	Algorithm	Algorithm Description						
		<p>Points to Note:</p> <p>The system does the following on the termination request date when you execute the C1-DPMON batch:</p> <ul style="list-style-type: none"> >> Changes the status of the delinquency process to the status (for example, Pending Termination) which is specified in the Delinquency Process Pending Termination Status attribute of the delinquency process preference. >> Updates the status reason of the policies or individual memberships billed to the delinquent person or account. <p>The system considers the delinquency process preference which is specified in the Delinquency Process Field Mapping option type of the DELINPROC feature configuration. For more information about the feature configuration, refer to the DELINPROC Feature Configuration section.</p> <p>This algorithm calculates the termination request date based on the termination trigger process and termination wait days (required if the Termination Trigger Process field is set to After X Days). The system derives the value of the Trigger Termination Process field and termination wait days from the delinquency process type or using an algorithm attached to the Termination and Reinstatement Configuration Derivation system event depending on the termination reinstatement configuration source that is specified in the delinquency process type.</p> <p>The following table explains how the termination request date is calculated for the delinquency event when the trigger termination process is set to:</p> <table border="1" data-bbox="751 1528 1412 1785"> <thead> <tr> <th data-bbox="760 1535 1040 1612">Trigger Termination Process is set to...</th> <th data-bbox="1044 1535 1404 1612">Termination Request Date is set to...</th> </tr> </thead> <tbody> <tr> <td data-bbox="760 1617 1040 1703">Immediate</td> <td data-bbox="1044 1617 1404 1703">Send Termination Request Event Trigger Date</td> </tr> <tr> <td data-bbox="760 1707 1040 1785">Month End</td> <td data-bbox="1044 1707 1404 1785">Last Day of Send Termination Request Event Trigger Month</td> </tr> </tbody> </table>	Trigger Termination Process is set to...	Termination Request Date is set to...	Immediate	Send Termination Request Event Trigger Date	Month End	Last Day of Send Termination Request Event Trigger Month
Trigger Termination Process is set to...	Termination Request Date is set to...							
Immediate	Send Termination Request Event Trigger Date							
Month End	Last Day of Send Termination Request Event Trigger Month							

Action	Algorithm	Algorithm Description	
		After X Days	Send Termination Request Event Trigger Date + Termination Wait Days
		<p>Note: The Send Termination Request event trigger date is calculated when you execute the C1-DPEVL batch.</p>	
		<p>Finally, this algorithm stores the termination request date of each delinquency process corresponding to the record in the C1_DELIN_PROC table.</p>	
Send Cancellation Request	C1-DLEVMBCNC	<p>This algorithm is used to update the status reason of an individual membership when its cancellation process is initiated through a delinquency process. Once the status reason of an individual membership is updated, this algorithm creates a log entry for the respective membership in the delinquency process.</p> <p>This algorithm is invoked when the cancellation event is triggered manually from the user interface or automatically through the C1-DPMON batch.</p> <p>You must use this algorithm in a delinquency process type using which you want to create a delinquency process to initiate cancellation request for an individual membership through an outbound message. It should be attached to the delinquency event type which requests to cancel the individual memberships billed to the delinquent account.</p> <div data-bbox="750 1243 1422 1713" style="border: 1px solid black; padding: 5px;"> <p>Points to Note:</p> <p>The system derives the active individual memberships using an algorithm which is attached to the Memberships Derivation system event of the delinquency process type.</p> <p>While deriving an active individual membership, the system considers the status code which is specified in the Membership Active Status option type of the C1-ASOBLNG feature configuration. For more information about the feature configuration, refer to the C1-ASOBLNG Feature Configuration section.</p> </div> <p>The system considers the status reason which is specified in the Awaiting Membership Cancellation Reason attribute of the delinquency process preference.</p>	

Action	Algorithm	Algorithm Description
		<p>Before updating the status reason of the individual membership, the system validates whether the status reason is defined for the Active status of the C1-IndMembership business object.</p> <div data-bbox="743 407 1416 646" style="border: 1px solid black; padding: 5px;"> <p>Note: The system considers the delinquency process preference which is specified in the Delinquency Process Field Mapping option type of the DELINPROC feature configuration. For more information about the feature configuration, refer to the DELINPROC Feature Configuration section.</p> </div>
Apply Advance Deposit	C1-APPADVDEP	<p>This algorithm transfers the required payment amount from the advance deposit contract to the delinquent bills by creating adjustments. It checks whether the delinquency process is created at the parent customer, bill group, or at the parent customer's or bill groups' account level. It then accordingly derives the parent customer and checks whether the advance deposit level of the parent customer is set to PG or POLI. If the advance deposit level of the parent customer is set to PG, the system derives the advance deposit account and advance deposit contract using the Advance Deposit Account Relationship Type and Advance Deposit Contract Type attribute, respectively, of the delinquency process preference. It then derives the sum of all the unmatched payments on the advance deposit contract. However, if the advance deposit level of the parent customer is set to POLI, the system derives the advance deposit account linked to the policy and the advance deposit contract using the Advance Deposit Contract Type attribute of the delinquency process preference. Once the advance deposit account and contract are derived for each policy, the system derives the sum of all the unmatched payments on the advance deposit contract for the respective policy.</p> <div data-bbox="743 1549 1416 1789" style="border: 1px solid black; padding: 5px;"> <p>Note: The system considers the delinquency process preference which is specified in the Delinquency Process Field Mapping option type of the DELINPROC feature configuration. For more information about the feature configuration, refer to the DELINPROC Feature Configuration section.</p> </div>

Action	Algorithm	Algorithm Description
		<p>It creates debit adjustments against the unmatched payments on the advance deposit contract and a single credit adjustment of the equivalent amount. The debit adjustments are created using the given debit adjustment type and a credit adjustment is created using the given credit adjustment type.</p> <p>The debit adjustments are matched against the payment segments and their match event status is set to Balanced. The system then creates an offset request using the given offset request type and includes the credit adjustment and delinquent bills in the offset request. It then automatically submits the offset request and creates the offset request adjustments against the credit adjustment and delinquent bills' bill segments. The offset request adjustments which are created against the bill segments are matched to the bill segments and their match event status is set to Balanced. The offset request adjustment which is created against the credit adjustment is matched to the credit adjustment and their match event status is set to Balanced. Finally, the status of the offset request is set to Processed. For more information, refer to the Settling Unpaid Dues Against Advance Deposit through a Delinquency Process section.</p> <p>This algorithm also derives the preferred contact method for the customer contact. While deriving the preferred contact method, this algorithm first fetches the main customer's bill route type on the advance deposit account and then fetches the bill routing method of the bill route type. Once the bill routing method is derived, the system considers the preference which is specified in the Routing Method – Contact Method Mapping option type of the DELINPROC feature configuration. For more information about the feature configuration, refer to the DELINPROC Feature Configuration section.</p> <p>It then checks whether any preferred contact method is mapped to the bill routing method in the preference. If the preferred contact method is mapped to the bill routing method in the preference, the system stamps it on the customer contact. However, if the preferred contact method is not mapped to the bill routing method in the preference, the system considers the default contact method as the preferred contact method and stamps it on the customer contact.</p>

Action	Algorithm	Algorithm Description
		<p>This algorithm creates a customer contact depending on whether the advance deposit is fully or partially utilized to pay the delinquent bills. The customer contact is created for the person or for the main customer of the account depending on whether the delinquency process is created at the person or account level. In addition, the system stamps the delinquency process ID as a characteristic using the given characteristic type on the customer contact and on the advance deposit payments (used for offsetting).</p> <div data-bbox="748 625 1421 703" style="border: 1px solid black; padding: 5px;"> <p>Note: This algorithm is only applicable for the fully insured group business.</p> </div> <p>This algorithm contains the following parameters:</p> <ul style="list-style-type: none"> • Debit Adjustment Type – Used to specify the adjustment type using which you want to create the debit adjustments against the payments on the advance deposit contract. • Credit Adjustment Type – Used to specify the adjustment type using which you want to create a credit adjustment that will be used to offset the overdue bills for which the delinquency process is created. • Offset Request Type – Used to specify the offset request type using which you want to create the offset request to offset the credit adjustment against the delinquent bills. Here, you must specify an offset request type where the offset category is set to Advance Deposit. • Customer Contact Type For Partial Usage Of Deposit – Used to indicate the customer contact type using which you want to create the customer contact when the advance deposit amount is partially utilized to pay the delinquent bills. Here, you must specify a customer contact type with the appropriate letter template depending on the type of letter you want to send to the customer. • Customer Contact Class For Partial Usage Of Deposit – Used to indicate the customer contact class to which the customer contact type specified in the Customer Contact Type For Partial Usage Of Deposit parameter belongs.

Action	Algorithm	Algorithm Description
		<ul style="list-style-type: none"> • Customer Contact Type For Depletion Of Deposit – Used to indicate the customer contact type using which you want to create the customer contact when the advance deposit amount is fully utilized to pay the delinquent bills. Here, you must specify a customer contact type with the appropriate letter template depending on the type of letter you want to send to the customer. • Customer Contact Class For Depletion Of Deposit – Used to indicate the customer contact class to which the customer contact type specified in the Customer Contact Type For Depletion Of Deposit parameter belongs. • Default Contact Method – Used to indicate the contact method that you want to use when the preferred contact method cannot be derived for the routing method from the preference. • Delinquency Process Characteristic Type – Used to indicate the characteristic type using which you want to store the delinquency process ID on the customer contact and payments. Here, you must specify a characteristic type where the characteristic entity is set to Customer Contact and Payment. <p>All these parameters are mandatory.</p>

Delinquency Event Completion

The following table lists the secondary actions that you can perform using the respective algorithm shipped with the product:

Action	Algorithm	Algorithm Description
Change the Delinquency Process Status	C1-TRNTERTRG	<p>This algorithm updates the status of the delinquency process on the termination request date. It is invoked in the C1-DPMON batch. The system considers the status which is specified in the Delinquency Process Pending Termination Status attribute of the delinquency process preference.</p> <div style="border: 1px solid black; padding: 5px;"> <p>Note: The system considers the delinquency process preference which is specified in the Delinquency Process Field Mapping option type of the DELINPROC feature configuration. For more information about the feature configuration, refer to the DELINPROC Feature Configuration section.</p> </div> <p>You must use this algorithm in a delinquency process type using which you want to create a delinquency process to initiate termination request for a policy or individual membership through an outbound message. It should be attached to the delinquency event type which requests to terminate the following:</p> <ul style="list-style-type: none"> • Policies billed to the delinquent person or account • Individual memberships billed to the delinquent account
Recalculate Trigger Dates Based on Document Mail Date	C1-RCLTRGDMD	<p>This algorithm fetches the customer contact which is created in the delinquency event and then retrieves its document mail date. If multiple customer contacts are created in the delinquency event, the system derives the latest document mail date. It then calculates the additional days required to deliver the document to the individual customer by finding the difference between the document mail date and the customer contact date.</p> <p>Once the additional days are derived, the system adds the additional days to the remaining events' trigger date and thereby derives the new trigger date for each remaining event in the delinquency process. If the Consider Work Days option is selected in the respective delinquency process type, the system ensures that the new trigger date falls on a business day and not on a holiday.</p> <p>This algorithm is invoked in the C1-DPMON batch. You can use this algorithm in a delinquency process type using which you want to create a delinquency process to send debt collection letters. It should be attached to the delinquency event type which sends a warning or dunning letter.</p>

Action	Algorithm	Algorithm Description
		<p>This algorithm contains the following parameters:</p> <ul style="list-style-type: none"> • Document Composition Date Characteristic Type – Used to indicate the characteristic type which stores the document mail date on the customer contact. This parameter is mandatory. Here, you must specify a characteristic type where the characteristic entity is set to Customer Contact. • Evaluate Document Mail Date For Primary Contact Only – Used to indicate whether the document mail date should be derived only from the delinquent account’s main customer’s customer contact. The valid values are Y and N. If you set the parameter value to N, the system derives the document mail date from all persons’ customer contacts who are associated with the delinquent account and then considers the latest document mail date. This parameter is optional. If you do not specify the parameter value, by default, it is set to N.

Delinquency Process Type

Once you create the delinquency event types, you need to create the required delinquency process types using which you want the system to create a delinquency process for an account. You can design various delinquency process types which trigger different types of events for the different types of insurance coverages. For example, you can define different delinquency process types for the following:

- Active health plan coverages
- Inactive health plan coverages
- Medical health plan coverages
- Non-medical health plan coverages
- Subsidized health plan coverages
- Non-effectuated health plan coverages

You can then configure the delinquency process type for each type of insurance coverage through the delinquency process type preference. The system considers the delinquency process type preference which is specified in the **Delinquency Process Type Field Mapping – Individual** parameter of the **C1-DLNQMNRL** algorithm (which is attached to the delinquency control).

A new maintenance object named **C1-DLQ-PRC-T**, a new business object named **C1-DelinProcType**, and a new screen named **Delinquency Process Type** are introduced in this release. The **Delinquency Process Type** screen enables you to create, edit, delete, and copy a delinquency process type. While creating a delinquency process type, you can specify the following:

- Basic details for the delinquency process type
- Details for the delinquency event trigger date calculation
- Types of delinquency events that you want to trigger in a delinquency process which is created using the delinquency process type

- Algorithms that you want to attach to the delinquency process type
- Details for the grace period calculation
- Details for the termination request (outbound message) trigger process
- Details for the termination date calculation
- Priority for each termination date rule
- Details for the trigger date recalculation when a delinquency process which is on hold is released and when a delinquency process which is canceled is resumed
- Actions for which you want to configure the approval process

Delinquency Process Type Details

While creating a delinquency process type, you need to specify the following details:

- The entity type (for example, **Account** or **Person**) for which the delinquency process type is applicable. Note that the account level delinquency process is applicable for both the fully insured individual and fully insured group businesses whereas the person level delinquency process is applicable only for the fully insured group business.
- The business object using which the delinquency process should be created in the system.

Note: A new maintenance object named **C1-DLN-PROC** and a new business object named **C1-DelinquencyProcess** are introduced in this release.

- The type of object (for example, bill) for which you want to manage collections through the delinquency process. Note that the system enables you select only those objects as the collection objects for which the **Eligible for Delinquency (Y/N)** option type is selected in the respective maintenance object. At present, the **Eligible for Delinquency (Y/N)** option type is only selected for the **BILL** maintenance object.
- Whether the termination process should be automatically initiated for the following:
 - Policies billed to the delinquent person or account
 - Individual memberships billed to the delinquent account
- Whether the reinstatement process should be automatically initiated for the policies billed to the delinquent person or account. The automatic reinstatement is applicable only for the fully insured group business.
- The way in which you want to trigger the delinquency events in the delinquency process. The system supports the following ways:
 - **Automatic** - Used when you want the system to automatically trigger delinquency events on the trigger date through the **C1-DPMON** batch. The trigger dates are calculated for the delinquency events when you execute the **C1-DPEVL** batch. The trigger dates of the delinquency events are calculated using the given trigger date calculation details, such as trigger date calculation mode, wait period, etc. If you set the delinquency event trigger mode to **Automatic**, the system does not allow you to trigger the delinquency events manually from the user interface. The **Trigger** icon corresponding to a delinquency event in the **Delinquency Process** screen is disabled.

- **Manual** - Used when you want to manually trigger delinquency events in the delinquency process at the desired interval. In this case, the trigger dates are not calculated for the delinquency events. The **Trigger** icon appears corresponding to a delinquency event in the **Delinquency Process** screen. You can trigger a delinquency event by clicking the **Trigger** icon corresponding to the delinquency event in any order irrespective of its sequence in the delinquency process. If you set the delinquency event trigger mode to **Manual**, the system does not automatically trigger delinquency events on the trigger date through the **C1-DPMON** batch.
- **Manual or Automatic** – Used when you want the system to automatically trigger delinquency events on the trigger date through the **C1-DPMON** batch. The trigger dates are calculated for the delinquency events when you execute the **C1-DPEVL** batch. The trigger dates of the delinquency events are calculated using the given trigger date calculation details, such as trigger date calculation mode, wait period, etc. Here, you also have an option to manually trigger a delinquency event in the delinquency process whenever required before the trigger date to expedite the collections process. The **Trigger** icon appears corresponding to a delinquency event in the **Delinquency Process** screen. You can trigger a delinquency event by clicking the **Trigger** icon corresponding to the delinquency event. Ideally, you must trigger a delinquency event in the ascending order of its sequence. However, if you manually trigger a delinquency event in any order, all the delinquency events in the **Pending** status before the manually triggered delinquency event will be skipped in the delinquency process. The status of such delinquency events is changed to **Skipped**.

Delinquency Event Trigger Date Calculation Details

While creating a delinquency process type, you need to specify the following details:

- Whether you want to delay a delinquency event for a specific period that can be specified either in days or months
- Whether the trigger date of a delinquency event should fall on a business day
- The method in which you want to calculate the trigger date for the delinquency events. The system supports the following methods:
 - **Latest Bill Due Date with Grace Period** – Used when you want to derive trigger date using the latest bill due date and grace period. The system considers the latest bill due date as the grace start date. It then derives the grace end date using the grace start date and grace period (in days). Once the grace start and end dates are derived, the system calculates the trigger date based on the trigger period and wait period configuration of the delinquency event type as shown below:

Trigger Period	Wait Period	Wait Days	Trigger Date is set to...
After Grace End Date	Wait Days	10	Grace End Date + 10 days
Before Grace End Date	Wait Days	5	Grace End Date - 5 days

Points to Note:

The system supports only wait days when the trigger date calculation mode is set to **Latest Bill Due Date with Grace Period**.

The system derives the wait days from the delinquency process type or using an algorithm attached to the **Wait Days Derivation** system event depending on the wait period source which is specified in the delinquency process type.

The system derives the grace period from the delinquency process type or using an algorithm attached to the **Grace Period Derivation** system event depending on the grace period source which is specified in the delinquency process type.

- **Paid Through Date** – Used when you want to derive trigger date using the paid through date of the account and the wait period. The system calculates the trigger date based on the wait period configuration of the delinquency event type as shown below:

Wait Period	Wait Days	Wait Months	Day of Month	Trigger Month is set to...	Trigger Date is set to...
Monthly Cyclic	-	1	5	Paid Through Date + 1 Month	5th of Trigger Month
Wait Days	5	-	-	-	Paid Through Date + 5 days

If the wait period is set to **Monthly Cyclic**, you need to specify the wait months and the day of the month for each delinquency event type in the delinquency process type. However, if the wait period is set to Wait Days, you need to specify the wait days for each delinquency event type in the delinquency process type.

Note: The system derives the wait days or wait month and the day of the month from the delinquency process type or using an algorithm attached to the **Wait Days Derivation** system event depending on the wait period source which is specified in the delinquency process type.

If the calculated trigger date of an event is past dated, the system enables you to move the trigger date of the event to the next month. If the new trigger date is also past dated, the system keeps moving the trigger date by one month until it is a future date. You can avail this functionality for the delinquency events by selecting the **Move Past Dated Event To Future** option corresponding to the respective delinquency event type in the delinquency process type. Note that if the trigger date of an event is moved to a future date, the trigger date of the subsequent events in the delinquency process is also moved accordingly.

This is because the system is designed to trigger only one event at a time in a month. While calculating the new trigger date, the system ensures that the delinquency events are triggered in the given sequence.

Note: You can enable the **Move Past Dated Event To Future** option only when the wait period is set to Monthly Cyclic.

- **Usage Basis** – Used when you want to derive trigger date using either of the following dates:
 - **Delinquency Process Creation Date** – Used when you want to derive trigger date using the delinquency process creation date and the wait period. The system calculates the trigger date based on the wait period configuration of the delinquency event type as shown below:

Wait Period	Wait Days	Wait Months	Day of Month	Trigger Month is set to...	Trigger Date is set to...
Monthly Cyclic	-	1	5	Delinquency Process Creation Date + 1 Month	5th of Trigger Month
Wait Days	15	-	-	-	Delinquency Process Creation Date + 15 days

If the wait period is set to **Monthly Cyclic**, you need to specify the wait months and the day of the month for each delinquency event type in the delinquency process type. However, if the wait period is set to **Wait Days**, you need to specify the wait days for each delinquency event type in the delinquency process type.

Note: The system derives the wait days or wait month and the day of the month from the delinquency process type or using an algorithm attached to the **Wait Days Derivation** system event depending on the wait period source which is specified in the delinquency process type.

If the calculated trigger date of an event is past dated, the system enables you to move the trigger date of the event to the next month. If the new trigger date is also past dated, the system keeps moving the trigger date by one month until it is a future date. You can avail this functionality for the delinquency events by selecting the **Move Past Dated Event To Future** option corresponding to the respective delinquency event type in the delinquency process type. Note that if the trigger date of an event is moved to a future date, the trigger date of the subsequent events in the delinquency process is also moved accordingly. This is because the system is designed to trigger only one event at a time in a month. While calculating the new trigger date, the system ensures that the delinquency events are triggered in the given sequence.

Note: You can enable the **Move Past Dated Event To Future** option only when the wait period is set to **Monthly Cyclic**.

- **Grace Start Date** - Used when you want to derive trigger date using the grace start date and the wait period. You can set the grace start date to either of the following - **Delinquency Process Creation Date**, **Latest Bill Due Date**, or **X Days after Latest Bill Due Date**. The system calculates the trigger date based on the wait period configuration of the delinquency event type as shown below:

Grace Start Date Basis	Wait Period	Wait Days	Wait Months	Day of Month	Add Days to Due Date	Trigger Month is set to...	Trigger Date is set to...
Delinquency Process Creation Date	Monthly Cyclic	-	1	5	-	Grace Start Date (i.e. Delinquency Process Creation Date) + 1 Month	5th of Trigger Month
Delinquency Process Creation Date	Wait Days	15	-	-	-	-	Grace Start Date (i.e. Delinquency Process Creation Date) + 15 days
Latest Bill Due Date	Monthly Cyclic	-	2	7	-	Grace Start Date (i.e. Latest Bill Due Date) + 2 Month	7th of Trigger Month
Latest Bill Due Date	Wait Days	20	-	-	-	-	Grace Start Date (i.e. Latest Bill Due Date) + 20 days
X Days after Latest Bill Due Date	Monthly Cyclic	-	1	10	10	Grace Start Date (i.e. Latest Bill Due Date + 10 days) + 1 Month	10th of Trigger Month
X Days after Latest Bill Due Date	Wait Days	20	-	-	10	-	Grace Start Date (i.e. Latest Bill Due Date + 10 days) + 20 days

If the wait period is set to **Monthly Cyclic**, you need to specify the wait months and the day of the month for each delinquency event type in the delinquency process type. However, if the wait period is set to **Wait Days**, you need to specify the wait days for each delinquency event type in the delinquency process type.

Note: The system derives the wait days or wait month and the day of the month from the delinquency process type or using an algorithm attached to the **Wait Days Derivation** system event depending on the wait period source which is specified in the delinquency process type.

If the calculated trigger date of an event is past dated, the system enables you to move the trigger date of the event to the next month. If the new trigger date is also past dated, the system keeps moving the trigger date by one month until it is a future date. You can avail this functionality for the delinquency events by selecting the **Move Past Dated Event To Future** option corresponding to the respective delinquency event type in the delinquency process type. Note that if the trigger date of an event is moved to a future date, the trigger date of the subsequent events in the delinquency process is also moved accordingly. This is because the system is designed to trigger only one event at a time in a month. While calculating the new trigger date, the system ensures that the delinquency events are triggered in the given sequence.

Note: You can enable the **Move Past Dated Event To Future** option only when the wait period is set to **Monthly Cyclic**.

- **Latest Bill Date** - Used when you want to derive trigger date using the latest bill date and the wait period. The system calculates the trigger date based on the wait period configuration of the delinquency event type as shown below:

Wait Period	Wait Days	Wait Months	Day of Month	Trigger Month is set to...	Trigger Date is set to...
Monthly Cyclic	-	1	5	Latest Bill Date + 1 Month	5th of Trigger Month
Wait Days	15	-	-	-	Latest Bill Date + 15 days

If the wait period is set to **Monthly Cyclic**, you need to specify the wait months and the day of the month for each delinquency event type in the delinquency process type. However, if the wait period is set to **Wait Days**, you need to specify the wait days for each delinquency event type in the delinquency process type.

Note: The system derives the wait days or wait month and the day of the month from the delinquency process type or using an algorithm attached to the **Wait Days Derivation** system event depending on the wait period source which is specified in the delinquency process type.

If the calculated trigger date of an event is past dated, the system enables you to move the trigger date of the event to the next month. If the new trigger date is also past dated, the system keeps moving the trigger date by one month until it is a future date. You can avail this functionality for the delinquency events by selecting the **Move Past Dated Event To Future** option corresponding to the respective delinquency event type in the delinquency process type. Note that if the trigger date of an event is moved to a future date, the trigger date of the subsequent events in the delinquency process is also moved accordingly.

This is because the system is designed to trigger only one event at a time in a month. While calculating the new trigger date, the system ensures that the delinquency events are triggered in the given sequence.

Note: You can enable the **Move Past Dated Event To Future** option only when the wait period is set to **Monthly Cyclic**.

- **Latest Bill Due Date** – Used when you want to derive trigger date using the latest bill due date and the wait period. The system calculates the trigger date based on the wait period configuration of the delinquency event type as shown below:

Wait Period	Wait Days	Wait Months	Day of Month	Trigger Month is set to...	Trigger Date is set to...
Monthly Cyclic	-	2	15	Latest Bill Due Date + 2 Month	15th of Trigger Month
Wait Days	25	-	-	-	Latest Bill Due Date + 25 days

If the wait period is set to **Monthly Cyclic**, you need to specify the wait months and the day of the month for each delinquency event type in the delinquency process type. However, if the wait period is set to **Wait Days**, you need to specify the wait days for each delinquency event type in the delinquency process type.

Note: The system derives the wait days or wait month and the day of the month from the delinquency process type or using an algorithm attached to the **Wait Days Derivation** system event depending on the wait period source which is specified in the delinquency process type.

If the calculated trigger date of an event is past dated, the system enables you to move the trigger date of the event to the next month. If the new trigger date is also past dated, the system keeps moving the trigger date by one month until it is a future date. You can avail this functionality for the delinquency events by selecting the **Move Past Dated Event To Future** option corresponding to the respective delinquency event type in the delinquency process type. Note that if the trigger date of an event is moved to a future date, the trigger date of the subsequent events in the delinquency process is also moved accordingly. This is because the system is designed to trigger only one event at a time in a month. While calculating the new trigger date, the system ensures that the delinquency events are triggered in the given sequence.

Note: You can enable the **Move Past Dated Event To Future** option only when the wait period is set to **Monthly Cyclic**.

Delinquency Event Types

While adding the delinquency event types in a delinquency process type, you need to specify the following:

- The delinquency event types using which you want to create the delinquency events in a delinquency process which is created using the delinquency process type
- The order in which you want to create the delinquency event using the delinquency event type
- The time period (i.e. either before or after grace end date) during which you want to trigger the delinquency events when the trigger date calculation mode is set to **Latest Bill Due Date with Grace Period**
- Whether you want the system to derive the wait days, wait months, and day of month during the trigger date calculation from the delinquency process type or through an algorithm
- The number of days for which you want to delay the delinquency event in a delinquency process
- The number of months for which you want to delay the delinquency event in a delinquency process
- The day of the month when you want to trigger the delinquency event of a delinquency process
- Whether you want the system to recalculate the trigger date of a delinquency event if it is a past dated until it is a future date
- Whether you want to preview a delinquency event which helps the relationship manager to inform the customer in advance that he will receive the To Do notification or customer contact within the specified number of days
- The number of days before the trigger date when you want to schedule a preview of a delinquency event
- Whether you want the system to skip the delinquency event in a delinquency process which is past dated
- Whether you want the system to check whether a delinquency event type is eligible for an account and then accordingly create the delinquency event using the delinquency event type

Points To Note:

The list includes only those rule types where the rule type usage is set to **Delinquency Event Type Eligibility**.

The membership level notification is not applicable for the fully insured individual business.

Delinquency Process Type Algorithms

While creating a delinquency process type, you need to attach the following algorithms to the respective system event:

System Event	Algorithm	Algorithm Description
Calculate Unpaid and Original Amounts	C1-CUAOAB	<p>This algorithm derives the original and unpaid amounts for each overdue bill of the person or account whose debt is monitored. It invokes an algorithm created using the C1-OI-BI-AMT algorithm type attached to the Determine Open Item Bill Amounts system event in the installation options to calculate the original and unpaid amounts for a bill.</p> <div style="border: 1px solid black; padding: 5px;"> <p>Points to Note:</p> <p>This algorithm is required when the collection object in the delinquency process type is set to Bill.</p> <p>The C1-CUAOAB algorithm is invoked from the following algorithms - C1-DLNQMNRL, C1-RODPDUEPX.</p> </div>
Cancel Criteria	C1-CANDELPRC	<p>This algorithm cancels the delinquency process when either of the following condition is met:</p> <ul style="list-style-type: none"> • Payments or adjustments are made against the overdue bills, for which the delinquency process is created, and as a result the unpaid balance of the overdue bills is within the tolerance limit • Collection class of the account, for which the delinquency process is created, is changed • Voluntary termination is received for an individual membership (which is billed to the delinquent account) with a termination date that is earlier than or equal to the calculated termination date • Voluntary cancellation is received for an individual membership (which is billed to the delinquent account) • Individual membership is canceled through an inbound message but its corresponding delinquency process is still in the Delinquency In Progress status

System Event	Algorithm	Algorithm Description
		<p>It contains the following parameters:</p> <ul style="list-style-type: none"> • Tolerance Percentage - Used to specify the unpaid percentage value (i.e. unpaid amount with respect to original amount). The system cancels the delinquency process when the unpaid percentage of the overdue bills is within the tolerance limit. This parameter is required. • Tolerance Amount - Used to specify the flat unpaid amount. The system cancels the delinquency process when the unpaid amount of the overdue bills is within the tolerance limit. This parameter is required. • Tolerance Amount and Percentage Required (Y / N) - Used to indicate whether you want to consider both the unpaid amount and unpaid percentage to check whether the delinquency process should be canceled or not. The valid values are: <ul style="list-style-type: none"> ○ Y - If you set the value to Y, the system considers both the unpaid amount and the unpaid percentage to check whether the delinquency process should be canceled or not. ○ N - If you set the value to N, the system considers either the unpaid amount or the unpaid percentage to check whether the delinquency process should be canceled or not. <p>This parameter is mandatory.</p> • Include On Account Payments In Threshold Evaluation - (Y/N) - Used to indicate whether you want to consider the payments made against the on account contracts of the account while calculating the unpaid amount. The valid values are Y and N. • On Account Payment Contract Type (Comma separated valid values) - Used to indicate the on account contract types that you want to consider while determining the total unpaid amount. You can specify maximum twenty comma-separated values for this parameter. <p>This parameter is only required when the Include On Account Payments In Threshold Evaluation - (Y/N) parameter is set to Y.</p>

System Event	Algorithm	Algorithm Description
		<ul style="list-style-type: none"> • Use Current Revenue Period Billed For Latest Overdue Due Date In Threshold Evaluation - (Y/N) – Used to indicate whether the system should calculate the unpaid percentage or amount based on the total amount billed for the coverage period or based on the total amount billed for the overdue bills. The valid values are: <ul style="list-style-type: none"> ○ Y – Used when you want to calculate the unpaid percentage or amount based on the total amount billed for the coverage period. ○ N - Used when you want to calculate the unpaid percentage or amount based on the total amount billed for the overdue bills. <p>This parameter is optional. If you do not specify the parameter value, by default, it is set to N.</p> • Which Date Determines Age ('D' - Due Date or 'B' - Bill Date) - Used to indicate whether you want the system to determine the age of the invoice (i.e. bill) using the bill date or the due date. The valid values are D and B. This parameter is mandatory. • Only Process Bills with an Age >= This Value - Used to specify the number of days. The system considers an overdue bill of the delinquency process only when the invoice age is greater than or equal to the specified number of days. This parameter is optional. • Delinquency Process Cancel Reason for Unpaid Amount Reduction – Used to indicate the cancel reason that you want to use while canceling a delinquency process when the unpaid amount and/or percentage is within the tolerance limit. Here, you must specify a reason which is defined for the Canceled status of the C1-DelinquencyProcess business object in the Status Reason screen. <p>Delinquency Process Cancel Reason for Collection Class Change - Used to indicate the cancel reason that you want to use while canceling a delinquency process when the collection class of the account is changed. Here, you must specify a reason which is defined for the Canceled status of the C1-DelinquencyProcess business object in the Status Reason screen.</p>

System Event	Algorithm	Algorithm Description
		<ul style="list-style-type: none"> • Delinquency Process Cancel Reason for Voluntary Termination – Used to indicate the cancel reason that you want to use while canceling a delinquency process when an individual membership billed to the account is voluntarily terminated by the customer on or before the calculated termination date. Here, you must specify a reason which is defined for the Canceled status of the C1-DelinquencyProcess business object in the Status Reason screen. • Delinquency Process Cancel Reason for Membership Cancellation - Used to indicate the cancel reason that you want to use while canceling a delinquency process when an individual membership billed to the account is voluntarily canceled by the customer. Here, you must specify a reason which is defined for the Canceled status of the C1-DelinquencyProcess business object in the Status Reason screen. • Evaluate Customer Status when all Events are Completed – Used when you want the system to evaluate whether an individual membership billed to the delinquent account is voluntarily terminated or canceled by the customer even when all the events in the delinquency process are in the Completed status. The valid values are Y and N. This parameter is mandatory. If you set this parameter value to N, the system will evaluate whether an individual membership billed to the delinquent account is voluntarily terminated or canceled by the customer until all the events in the delinquency process are in the Completed status and not after all the events in the delinquency process are in the Completed status. <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>Note: The cancellation criteria only consider the bills that were due on or before the delinquency process was created. This is because the grace period is determined based on the latest due date. From a business perspective, the system should ensure that a customer is given sufficient grace period for the unpaid invoices.</p> </div>

System Event	Algorithm	Algorithm Description
Grace Period Derivation	C1-DERGRBR	<p>This algorithm is required when the Grace Period Source flag in the delinquency process type is set to Algorithm. It retrieves the business rules of the Delinquency Grace Period Business Rule category which are effective on the processing date when the C1-DPEVTCR algorithm is invoked. Note that the system considers only those effective business rules which are in the Active status.</p> <p>Once the effective business rules are identified, the system executes these business rules in the order of their priority. Once an individual membership meets the criteria defined in the business rule, the business rule is stamped against the individual membership. The system then derives the Grace Period (Days) flag from the business rule.</p> <p>The system then checks the grace period of each active individual membership and sets the grace period of the delinquency event to the value derived for the membership with the maximum grace days.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>Note: The system derives the individual memberships using an algorithm which is attached to the Memberships Derivation system event of the delinquency process type.</p> </div>
Hold Delinquency Process Criteria	C1-HLDDELPRC	<p>This algorithm holds a delinquency process when any of the following conditions are met:</p> <ul style="list-style-type: none"> • Postpone credit review date stamped on the delinquent account is later than the system date • One or more promise to pay are active for the delinquent account • A payment agreement request is active for the delinquent account <p>It is invoked by the C1-DPMON batch.</p> <p>If the postpone credit review date of the account is later than the system date, and/or if an active payment agreement request exists for the delinquent account, and/or if one or more promise to pay are active for the delinquent account, the system determines the hold end date. While determining the hold end date, it considers the date which is the latest of the following:</p> <ul style="list-style-type: none"> • Postpone credit review date • Latest schedule date specified in the payment agreement request

System Event	Algorithm	Algorithm Description
		<ul style="list-style-type: none"> • Latest schedule date specified among all the active promise to pay • Hold end date manually specified in the delinquency process <p>If the derived hold end date is later than the system date, then the system does the following:</p> <ul style="list-style-type: none"> • Sets the status of the delinquency process to the value specified in the Delinquency Process Hold Status attribute of the delinquency process preference. <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>Note: The system considers the delinquency process preference which is specified in the Delinquency Process Field Mapping option type of the DELINPROC feature configuration. For more information about the feature configuration, refer to the DELINPROC Feature Configuration section.</p> </div> <ul style="list-style-type: none"> • Sets the hold start date to a date which is earliest of the following: <ul style="list-style-type: none"> ○ If the postpone credit review date of the account is later than the system date, the system uses the processing date as the hold start date. ○ If an active payment agreement request exists for the delinquent account, the system uses the date when the payment agreement request was set to the Active status as the hold start date. ○ If one or more promise to pay are active for the delinquent account, the system uses the earliest schedule date among all the active promise to pay as the hold start date. • Sets the hold reason depending on whether the system considers either of the following as the hold end date: <ul style="list-style-type: none"> ○ Postpone credit review date ○ Latest schedule date specified in the payment agreement request ○ Latest schedule date specified among all the active promise to pay

System Event	Algorithm	Algorithm Description
		<ul style="list-style-type: none"> ○ Hold end date manually specified in the delinquency process ● Sets the mode of hold for the delinquency process to Systematic <p>This algorithm contains the following parameters:</p> <ul style="list-style-type: none"> ● Hold Request Reason – Used to indicate the reason that you want to use while holding a delinquency process. The system considers this reason when the hold end date is to set to the postpone credit review date of the account. Here, you must specify the reason which is defined for the On Hold status of the C1-DelinquencyProcess business object. ● Promise To Pay Reason - Used to indicate the reason that you want to use while holding a delinquency process. The system considers this reason when the hold end date is set to the latest schedule date specified in the payment agreement request. Here, you must specify the reason which is defined for the On Hold status of the C1-DelinquencyProcess business object. ● Payment Agreement Reason - Used to indicate the reason that you want to use while holding a delinquency process. The system considers this reason when the hold end date is set to the latest schedule date specified among all the active promise to pay. Here, you must specify the reason which is defined for the On Hold status of the C1-DelinquencyProcess business object. <p>All the parameters are mandatory.</p>

System Event	Algorithm	Algorithm Description
Memberships Derivation	C1-MEMBDLQ	<p>This algorithm fetches those individual memberships where details of the account, for which the delinquency process is created, are stored as the characteristics on the individual memberships. While fetching the individual memberships which are billed to delinquent account, the system considers the characteristic types which are specified in the Account Identifier Type Char Type and Account Identifier Type Char Value option types of the C1-ASOBLNG feature configuration.</p> <p>Once the individual memberships are derived, the system considers only those memberships where the status is set to the value specified in the Membership Active Status option type of the C1-ASOBLNG feature configuration.</p> <p>This algorithm then derives a preference where the preference category is set to Individual Membership. It checks whether the characteristic type specified in the Active Selection Characteristics attribute of the preference is defined on any of the derived active individual memberships.</p> <div data-bbox="743 1003 1419 1276" style="border: 1px solid black; padding: 5px;"> <p>Note: If a characteristic of the specified characteristic type exists on an individual membership, it means that the individual membership is created for the next year's coverage. However, if a characteristic of the specified characteristic type does not exist on an individual membership, it means that the individual membership is created for the current year's coverage.</p> </div> <p>If the account selection characteristic exists on the active individual membership, the system checks whether the characteristic's effective date is later than or equal to the membership start date. If the characteristic's effective date is later than or equal to the membership start date, the system checks whether the paid through date is stamped on the account. If the paid through date is stamped on the account, the system checks whether the membership start date is later than the paid through date. If the membership start date is later than the paid through date, the system derives the following:</p> <ul style="list-style-type: none"> • Sum of the following payments <ul style="list-style-type: none"> ○ Payment made against the on account contracts

System Event	Algorithm	Algorithm Description
		<ul style="list-style-type: none"> ○ Payments made against the coverage period that falls after the paid through date • Premium for the first coverage period of the individual membership (which is created for the next year’s coverage) <p>The system then compares the premium of the first coverage period against the sum of payments. If the payment is sufficient to pay the first coverage period premium, the system sets the Evaluate Guaranteed Availability flag of the individual membership to true.</p> <p>However, if the account selection characteristic is not defined on the individual membership or if the account selection characteristic’s effective date is earlier than the membership start date or if the paid through date is not stamped on the account or if the individual membership start date is earlier than or equal to the paid through date, or if the payment is not sufficient to pay the first coverage premium for the next year’s membership, then the system sets the Evaluate Guaranteed Availability flag of the individual membership to false.</p> <p>This algorithm contains the following parameter:</p> <ul style="list-style-type: none"> • On Account Payment Contract Types – Used to indicate the contract types whose contracts you want to consider while deriving the on account payments. This parameter is required. <p>Note: This algorithm is only applicable for the fully insured individual business.</p>

System Event	Algorithm	Algorithm Description
Policies Derivation	C1-POLDLQPA	<p>This algorithm derives the list of policies for a delinquency process. If the delinquency process is created at the account level, this algorithm derives the main customer of the account and then fetches all the policies where the main customer is added as a policy person in the policy.</p> <p>If the delinquency process is created at the person level, this algorithm derives all the policies where the person and its child persons are added as a policy person in the policy. Note that the algorithm considers only those child persons in the person’s hierarchy for whom the collection method in the respective collection class is set to Parental Delinquency.</p> <div data-bbox="743 743 1422 976" style="border: 1px solid black; padding: 5px;"> <p>Points to Note:</p> <p>If the status is passed through the calling algorithm, the policy list is filtered based on the given status.</p> <p>This algorithm is applicable only for the fully insured group business.</p> </div>

System Event	Algorithm	Algorithm Description
Release Delinquency Process Criteria	C1-REDELPRC	<p>This algorithm monitors a delinquency process which is on hold and determines whether the hold end date of the delinquency process should be updated or whether the hold on the delinquency process should be released. It is invoked by the C1-DPMON batch.</p> <p>If the postpone credit review date of the account is earlier than or equal to the system date, and if an active payment agreement request does not exist for the delinquent account, and if none of the promise to pay are active for the delinquent account, the system determines whether the hold end date is manually updated by the user. If the hold end date is manually updated by the user, the system sets the mode of hold for the delinquency process to Manual. However, if the hold end date is not updated manually by the user or if the manually specified hold end date is earlier than or equal to the system date, the system sets the status of the delinquency process to the value specified in the Delinquency Process Released Status attribute of the delinquency process preference.</p> <div data-bbox="743 995 1414 1234" style="border: 1px solid black; padding: 5px;"> <p>Note: The system considers the delinquency process preference which is specified in the Delinquency Process Field Mapping option type of the DELINPROC feature configuration. For more information about the feature configuration, refer to the DELINPROC Feature Configuration section.</p> </div> <p>If the postpone credit review date of the account is later than the system date, and/or if an active payment agreement request exists for the delinquent account, and/or if one or more promise to pay are active for the delinquent account, the system determines the hold end date. While determining the hold end date, it considers the date which is the latest of the following:</p> <ul style="list-style-type: none"> • Postpone credit review date • Latest schedule date specified in the payment agreement request • Latest schedule date specified among all the active promise to pay • Hold end date manually specified in the delinquency process

System Event	Algorithm	Algorithm Description
		<p>If the derived hold end date is later than the system date and if the derived hold end date is later than the existing hold end date, then the existing hold end date is updated to the derived hold end date. The hold reason is also updated depending on whether the system considers either of the following as the hold end date:</p> <ul style="list-style-type: none"> • Postpone credit review date • Latest schedule date specified in the payment agreement request • Latest schedule date specified among all the active promise to pay <p>In addition, the system sets the mode of hold for the delinquency process to Systematic.</p> <p>This algorithm contains the following parameters:</p> <ul style="list-style-type: none"> • Hold Request Reason – Used to indicate the reason that you want to use while holding a delinquency process. The system considers this reason when the hold end date is to set to the postpone credit review date of the account. Here, you must specify the reason which is defined for the On Hold status of the C1-DelinquencyProcess business object. • Promise To Pay Reason - Used to indicate the reason that you want to use while holding a delinquency process. The system considers this reason when the hold end date is set to the latest schedule date specified in the payment agreement request. Here, you must specify the reason which is defined for the On Hold status of the C1-DelinquencyProcess business object. • Payment Agreement Reason - Used to indicate the reason that you want to use while holding a delinquency process. The system considers this reason when the hold end date is set to the latest schedule date specified among all the active promise to pay. Here, you must specify the reason which is defined for the On Hold status of the C1-DelinquencyProcess business object. <p>All the parameters are mandatory.</p>
Send Member Level Notification Option Derivation	C1-DERMLNBR	<p>This algorithm is required when the Send Membership Level Notification Source field corresponding to the respective delinquency event type in the respective delinquency process type is set to Algorithm.</p>

System Event	Algorithm	Algorithm Description
		<p>If the delinquency process is created at the account level, this algorithm derives the main customer of the respective account. However, if the delinquency process is created at the person level, this algorithm derives the respective person. It then derives all the active individual memberships where the main customer is added as the member person.</p> <p>If no active individual membership is derived for the main customer, this algorithm derives all the active policies where the main customer or the person is added as the policy person in the respective policies. If the person is a bill group, the system considers the policies where the bill group is added as a policy person using the role which is specified in the Bill Group Policy Person Role option type of the C1-ASOBLNG feature configuration. However, if no active policy is derived for the bill group, this algorithm considers all the active policies where the parent customer of the bill group is added as a policy person using the role which is specified in the Parent Customer Policy Person Role option type of the C1-ASOBLNG feature configuration.</p> <p>However, if the person is a parent customer, the system considers all the active policies where the parent customer is added as a policy person using the role which is specified in the Parent Customer Policy Person Role option type of the C1-ASOBLNG feature configuration.</p> <div data-bbox="743 1207 1416 1598" style="border: 1px solid black; padding: 5px;"> <p>Points to Note:</p> <p>While deriving an active individual membership, the system considers the status code which is specified in the Membership Active Status option type of the C1-ASOBLNG feature configuration.</p> <p>Similarly, while deriving an active policy, the system considers the status code which is specified in the Policy Active Status option type of the C1-ASOBLNG feature configuration.</p> </div> <p>Once the list of active policies or individual memberships is derived, this algorithm retrieves the business rules of the Delinquency Event Attributes Business Rule category which are effective when the event is triggered manually from the user interface or automatically through the C1-DPMON batch.</p>

System Event	Algorithm	Algorithm Description
		<p>Note that the algorithm considers only those effective business rules which are in the Active status.</p> <p>Once the effective business rules are identified, this algorithm executes these business rules in the order of their priority. Once a policy or individual membership meets the criteria defined in the business rule, the system refers the business rule for the policy or individual membership. This algorithm then derives the value of the Send Membership Level Notification parameter from the business rule.</p> <p>It then checks whether the Send Membership Level Notification parameter is set to Send Membership Level Notification Required for at least one policy or individual membership billed to the delinquent customer or account. If so, the algorithm sets the Send Membership Level Notification parameter for the delinquency event to Send Membership Level Notification Required. However, if the Send Membership Level Notification parameter is set to Send Membership Level Notification Not Required for all the policies or individual memberships billed to the delinquent customer or account, the algorithm sets the Send Membership Level Notification parameter for the delinquency event to Send Membership Level Notification Not Required.</p>
Termination Date Rule Derivation	C1-DERTBR	<p>This algorithm is invoked by the C1-DETERMMDT algorithm when the event is triggered to send a termination request manually from the user interface or automatically through the C1-DPMON batch.</p> <p>If the delinquency process is created at the account level, this algorithm derives the main customer of the respective account. However, if the delinquency process is created at the person level, this algorithm derives the respective person. It then derives all the active individual memberships where the main customer is added as the member person.</p> <p>If no active individual membership is derived for the main customer, this algorithm derives all the active policies where the main customer or the person is added as the policy person in the respective policies. If the person is a bill group, the system considers the policies where the bill group is added as a policy person using the role which is specified in the Bill Group Policy Person Role option type of the C1-ASOBLNG feature configuration.</p>

System Event	Algorithm	Algorithm Description
		<p>However, if no active policy is derived for the bill group, this algorithm considers all the active policies where the parent customer of the bill group is added as a policy person using the role which is specified in the Parent Customer Policy Person Role option type of the C1-ASOBLNG feature configuration.</p> <p>However, if the person is a parent customer, the system considers all the active policies where the parent customer is added as a policy person using the role which is specified in the Parent Customer Policy Person Role option type of the C1-ASOBLNG feature configuration.</p> <div data-bbox="743 667 1416 1056" style="border: 1px solid black; padding: 5px;"> <p>Points to Note:</p> <p>While deriving an active individual membership, the system considers the status code which is specified in the Membership Active Status option type of the C1-ASOBLNG feature configuration.</p> <p>Similarly, while deriving an active policy, the system considers the status code which is specified in the Policy Active Status option type of the C1-ASOBLNG feature configuration.</p> </div> <p>Once the list of active policies or individual memberships is derived, this algorithm retrieves the business rules of the Delinquency Termination Date Rule Business Rule category which are effective on the trigger date. Note that the system considers only those effective business rules which are in the Active status.</p> <p>Once the effective business rules are identified, the system executes these business rules in the order of their priority.</p> <p>Once a policy or individual membership meets the criteria defined in the business rule, the system refers the business rule for the policy or individual membership. This algorithm then derives the value of the Termination Date Rule parameter from the business rule. For each active policy which is billed to the delinquent person or account, the system derives the value of the Termination Date Rule parameter. Similarly, for each active individual membership which is billed to the delinquent account, the system derives the value of the Termination Date Rule parameter.</p>

System Event	Algorithm	Algorithm Description
		<p>Among the termination date rules derived for different policies or individual memberships, this algorithm considers the termination date rule which is given the highest priority in the respective delinquency process type and uses it for further processing in the delinquency event.</p>
Termination and Reinstatement Configuration Derivation	C1-DERTROBR	<p>This algorithm is invoked by the C1-DETRTGD and C1-RECALCHLD algorithms when the Termination Reinstatement Configuration Source flag in the delinquency process type is set to Algorithm. It retrieves the business rules of the Delinquency Miscellaneous Options Business Rule category which are effective on the batch business date. Note that the system considers only those effective business rules which are in the Active status. Once the effective business rules are identified, the system executes these business rules in the order of their priority. Once an individual membership meets the criteria defined in the business rule, the business rule is stamped against the individual membership. The system then derives the value for the Trigger Termination Process flag and the termination wait days from the business rule.</p> <p>The system checks whether the Trigger Termination Process flag is set to Immediate for all the active individual memberships which are billed to the delinquent account. If so, the system sets the termination request date of the delinquency event to the termination event trigger date and the Trigger Termination Process flag of the delinquency event to Immediate.</p> <p>However, if the Trigger Termination Process flag is set to Month End or After X Days for any or all the active memberships, the system calculates the termination request date for each active membership and then considers the latest termination request date of the individual membership.</p> <p>The system sets the termination request date of the delinquency event to the latest termination request date of an active membership and sets the Trigger Termination Process flag of the delinquency event to the value derived for the membership with the latest termination request date.</p>

System Event	Algorithm	Algorithm Description
		<p>Note: The system derives the individual memberships using an algorithm which is attached to the Memberships Derivation system event of the delinquency process type.</p>
Trigger Date Recalculation Option Derivation	C1-DERRTDBR	<p>This algorithm is required when the Trigger Date Recalculation on Hold Release Source flag or Trigger Date Recalculation on Resume Source flag in the delinquency process type is set to Algorithm. It retrieves the business rules of the Delinquency Miscellaneous Options Business Rule category which are effective on the batch business date while executing the C1-DPMON batch. Note that the system considers only those effective business rules which are in the Active status.</p> <p>Once the effective business rules are identified, the system executes these business rules in the order of their priority. Once an individual membership meets the criteria defined in the business rule, the business rule is stamped against the individual membership. The system then derives the value for the Trigger Date Recalculation on Hold Release or Trigger Date Recalculation on Resume flag, respectively, from the business rule.</p> <p>For each active individual membership which is billed to the delinquent account, the system checks the value of the Trigger Date Recalculation on Hold Release flag while releasing the hold on the delinquency process. If the Trigger Date Recalculation on Hold Release flag is set to Yes for at least one active membership, the system sets the Trigger Date Recalculation on Hold Release flag of the delinquency event to Yes.</p> <p>However, if the Trigger Date Recalculation on Hold Release flag is set to No for all active memberships, the system sets the Trigger Date Recalculation on Hold Release flag of the delinquency event to No.</p> <p>Similarly, for each active individual membership which is billed to the delinquent account, the system checks the value of the Trigger Date Recalculation on Resume flag while resuming the delinquency process.</p> <p>If the Trigger Date Recalculation on Resume flag is set to Yes for at least one active membership, the system sets the Trigger Date Recalculation on Resume flag of the delinquency event to Yes.</p>

System Event	Algorithm	Algorithm Description
		<p>However, if the Trigger Date Recalculation on Resume flag is set to No for all active memberships, the system sets the Trigger Date Recalculation on Resume flag of the delinquency event to No.</p> <div style="border: 1px solid black; padding: 5px;"> <p>Note: The system derives the individual memberships using an algorithm which is attached to the Memberships Derivation system event of the delinquency process type.</p> </div>
Wait Days Derivation	C1-DERWDBR	<p>This algorithm is required when the Wait Period Source field corresponding to the respective delinquency event type in the respective delinquency process type is set to Algorithm.</p> <p>If the delinquency process is created at the account level, this algorithm derives the main customer of the respective account. However, if the delinquency process is created at the person level, this algorithm derives the respective person. It then derives all the active individual memberships where the main customer is added as the member person.</p> <p>If no active individual membership is derived for the main customer, this algorithm derives all the active policies where the main customer or the person is added as the policy person in the respective policies. If the person is a bill group, the system considers the policies where the bill group is added as a policy person using the role which is specified in the Bill Group Policy Person Role option type of the C1-ASOBLNG feature configuration. However, if no active policy is derived for the bill group, this algorithm considers all the active policies where the parent customer of the bill group is added as a policy person using the role which is specified in the Parent Customer Policy Person Role option type of the C1-ASOBLNG feature configuration.</p> <p>However, if the person is a parent customer, the system considers all the active policies where the parent customer is added as a policy person using the role which is specified in the Parent Customer Policy Person Role option type of the C1-ASOBLNG feature configuration.</p>

System Event	Algorithm	Algorithm Description
		<p>Points to Note:</p> <p>While deriving an active individual membership, the system considers the status code which is specified in the Membership Active Status option type of the C1-ASOBLNG feature configuration.</p> <p>Similarly, while deriving an active policy, the system considers the status code which is specified in the Policy Active Status option type of the C1-ASOBLNG feature configuration.</p> <p>Once the list of active policies or individual memberships is derived, this algorithm retrieves the business rules of the Delinquency Event Attributes Business Rule category which are effective on the batch business date while executing the C1-DPEVL batch. Note that the system considers only those effective business rules which are in the Active status.</p> <p>Once the effective business rules are identified, this algorithm executes these business rules in the order of their priority. Once a policy or individual membership meets the criteria defined in the business rule, the system refers the business rule for the policy or individual membership. This algorithm then derives the wait days or wait month and day of month from the business rule depending on the wait period that is configured in the respective delinquency process type.</p> <p>For each active policy which is billed to the delinquent person or account, the system derives the wait days or wait month and day of month. Similarly, for each active individual membership which is billed to the delinquent account, the system derives the wait days or wait month and day of month. Among the wait days or wait months derived for different policies or individual memberships, this algorithm considers the wait days or wait months which is maximum and uses it for further processing in the delinquency event.</p>

Grace Period Configuration

While creating a delinquency process type, you need to specify the following grace details using which the system calculates the trigger date for a delinquency event in the delinquency process.

- Whether you want the system to derive the grace period (in days) to calculate grace end date from the delinquency process type or through an algorithm

- The number of days of grace you want to offer from the grace start date
- Whether you want the system to set either of the following as the grace start date when the **Trigger Date Calculation Mode** flag is set to **Usage Basis** and the **Usage Basis** flag is set to **Grace Start Date**:
 - **Delinquency Process Creation Date** – Used when you want to set the grace start date to the date when the delinquency process is created in the system.
 - **Latest Bill Due Date** – Used when you want to set the grace start date to the latest due date of a bill when there are one or more overdue bills for which the delinquency process is initiated.
 - **X Days after Latest Bill Due Date** – Used when you want to set the grace start date after X number of days from the latest bill due date. You can specify the X number of days in the **Add Days To Due Date** field.

Termination and Reinstatement Configuration

While creating a delinquency process type, you need to specify the following details using which you want to calculate the termination request date:

- Whether you want the system to derive the value for the **Trigger Termination Process** flag and the termination wait days from the delinquency process type or through an algorithm
- Whether you want to send an outbound message to the enrollment system for terminating the individual memberships billed to the delinquent account immediately, at the end of the month when the termination event is triggered, or after X number of days (i.e. termination wait days)

Points to Note:

The **Allow Automatic Reinstatement**, **Maximum Automatic Reinstatement Count**, **Trigger Reinstatement Process**, **Reinstatement Wait Days**, **Reinstatement Threshold**, **Days Since Termination**, **Membership Notification Days Source**, and **Member Notification Days** fields are not applicable for the fully insured individual business.

Also, the **Termination Reasons For Automatic Reinstatement** section is not applicable for the fully insured individual business.

Termination Date Rule Priority

While creating a delinquency process type, you need to define the priority for each termination date rule based on the requirements. The system will then consider the termination date rule in the order of its priority. For example, if you have set the following priority in the DT1 delinquency process type:

Termination Date Rule	Priority
Latest Billed Coverage End Date	10
Latest Due Coverage End Date	20
X days After Paid Through Date	30
X Months After Paid Through Date	40
Month End of Grace Period Start	50
Month End of Termination Letter Creation	60
Month End of Termination Request Event	70
X days After Grace End Date	80
X days After Termination Request Event	90

Let us assume that the A1 account is used to bill the M1 and M2 individual memberships. Now, when the A1 account is delinquent, the system derives the **X days After Paid Through Date** terminate date rule for the M1 membership and the **X days After Grace End Date** termination date rule for the M2 membership using the business rules. The system then considers the termination date rule with the highest priority defined in the respective delinquency process type. In this case, the **X days After Paid Through Date** termination date rule has higher priority than the **X days After Grace End Date** termination date rule, and therefore the system sets the termination date rule of the delinquency event to **X days After Paid Through Date**.

Trigger Date Recalculation Configuration

The system enables you to recalculate trigger dates of the events when a delinquency process which is on hold is released. To enable this feature, you need to specify the following details while creating the respective delinquency process type:

- Whether you want the system to derive the value for the **Trigger Date Recalculation on Hold Release** flag from the delinquency process type or through an algorithm
- Whether you want to recalculate trigger dates of the events when a delinquency process which is on hold is released

Similarly, the system enables you to recalculate trigger dates of the events when a delinquency process which is canceled is resumed. To enable this feature, you need to specify the following details while creating the respective delinquency process type:

- Whether you want the system to derive the value for the **Trigger Date Recalculation on Resume** flag from the delinquency process type or through an algorithm
- Whether you want to recalculate trigger dates of the events when a delinquency process which is canceled is resumed

Approval Configuration

The system enables you to configure the approval process for various actions that you can perform for a delinquency process. For example, the system can initiate the approval process when a user manually holds a delinquency process. Note that you can configure the approval process only for any manual intervention in the delinquency process. To configure the approval process, you need to specify the following details while creating a delinquency process type:

- Whether the approval process must be initiated or not when a user performs certain actions
- The To Do type using which you want to create a To Do for the approver
- The To Do role which indicates the users for whom the To Do should be created
- The user actions for which you want to configure the approval process

You can configure the approval process for the following actions which can be manually performed by a user:

- Cancel
- Edit Hold
- Edit Trigger Date
- Hold
- Release Hold
- Request Reinstatement
- Skip Event
- Trigger Event

Delinquency Event Eligibility Evaluation and Trigger Date Calculation

After executing the **C1-ACDLQ** batch, you need to execute the **C1-DPEVL** batch. The **C1-DPEVL** batch is used to monitor or check whether there is any delinquency process in the status (for example, **Initiated**) which is specified in the **Delinquency Process Initiated Status** attribute of the delinquency process preference. Note that this batch considers the preference which is specified in the **Delinquency Process Field Mapping** option type of the **DELINPROC** feature configuration. For more information about the feature configuration, refer to the [DELINPROC Feature Configuration](#) section.

If there is a delinquency process in the **Initiated** status, the system checks whether its delinquency event trigger mode is set to **Automatic** or **Manual or Automatic** in the respective delinquency process type. If its delinquency event trigger mode is set to **Manual** in the respective delinquency process type, the batch does not consider the delinquency process for the delinquency event eligibility evaluation and trigger date calculation. However, if its delinquency event trigger mode is set to **Automatic** or **Manual or Automatic** in the respective delinquency process type, the system checks whether the trigger dates exist for the delinquency events. If the trigger dates exist for the delinquency events of a delinquency process, the batch does not consider the delinquency process for the delinquency event eligibility evaluation and trigger date calculation. However, if the trigger dates does not exist for the delinquency events of a delinquency process, the batch continues with the delinquency event eligibility evaluation for each delinquency event in the delinquency process.

While evaluating whether the delinquency event is eligible to be included in the delinquency process, the system checks whether the rule type is specified corresponding to the delinquency event type in the respective delinquency process type. If the rule type is specified corresponding to the delinquency event type, the system checks whether the delinquency entity type of the delinquency process is set to **Account** and the delinquency category is set to **Individual**. If so, the system derives the individual memberships which are billed to the delinquent account using an algorithm which is attached to the **Memberships Derivation** system event of the respective delinquency process type.

Once the individual memberships are derived for the account, the system derives the health plan for which the individual membership is availed. Once the health plan is derived for each individual membership, the system derives the rules created using the rule type (which is specified to corresponding to the delinquency event type) from the rules engine which are effective on the batch business date. The effective rules are executed in the order of their priority. If at least one individual membership satisfies the rule, the delinquency event is retained in the delinquency process. However, if none of the individual memberships satisfy the rule, the delinquency event is removed from the delinquency process.

Once the delinquency event eligibility evaluation is done for each delinquency event, the system does the following:

- Derives the wait days or wait month and the day of the month from the delinquency process type or using an algorithm attached to the **Wait Days Derivation** system event depending on the wait period source which is specified in the delinquency process type

- Calculates the trigger date for the delinquency event based on the trigger date calculation mode, usage basis, and wait period defined in the delinquency process type
- Sets the delinquency due date of the delinquency process to the latest bill due date

If the **Consider Work Days** option is selected in the respective delinquency process type, the system derives the work calendar of the division to which the account belongs. The system then checks whether the calculated trigger date falls on the working (i.e. business) day. If the calculated trigger date does not fall on the working (i.e. business) day, the system will accordingly add days to the calculated trigger date to ensure that it falls on the next working day.

If the calculated trigger date of a delinquency event is later than or equal to the batch business date, the status of the delinquency event is changed from **Pending Evaluation** to **Pending**. But, if the calculated trigger date of a delinquency event is earlier than the batch business date and if the **Skip Past Event** option is selected corresponding to the respective delinquency event type, the system sets the status of the delinquency event to **Skipped**. However, if the calculated trigger date of a delinquency event is earlier than the batch business date and if the **Skip Past Event** option is not selected corresponding to the respective delinquency event type, the system checks whether the **Move Past Dated Event to Future** option is selected corresponding to the respective delinquency event type.

Note: You can select the **Move Past Dated Event to Future** option only when the wait period is set to **Monthly Cyclic** in the respective delinquency process type.

If the **Move Past Dated Event to Future** option is selected corresponding to the respective delinquency event type, the system moves the trigger date by adding one month until the trigger date is later than or equal to the batch business date. But, if the **Move Past Dated Event to Future** option is not selected corresponding to the respective delinquency event type, the system makes no changes to the trigger date even if it is past dated. Finally, the system sets the status of the delinquency event to **Pending**.

Points to Note:

If the trigger date of an event is moved to a future date, the trigger date of the subsequent events in the delinquency process is also moved accordingly. This is because the system is designed to trigger only one event at a time in a month.

Usually, this batch changes the status of the delinquency event from **Pending Evaluation** to **Pending**. But, if the wait period in the respective delinquency process type is set to **Monthly Cyclic** and if the **Move Past Dated Event to Future** option is selected corresponding to the respective delinquency event type, the **C1-ADJTGDT** algorithm attached to the **Post-Processing** system event of the **C1-DPEVL** batch moves the past dated trigger date to a future date and changes the status of the delinquency event to **Pending**.

If the **Preview Required** option is not selected corresponding to the delinquency event type in the respective delinquency process type, the system sets the preview status of the delinquency event to **Not Required**. However, if the **Preview Required** option is selected corresponding to the delinquency event type in the respective delinquency process type, the system does the following:

- Derives the preview days specified corresponding to the delinquency event type in the delinquency process type
- Calculates the preview date (i.e. Trigger Date – Preview Days) for the delinquency event

- Sets the preview status of the delinquency event to **Pending**

The **C1-DPEVL** batch is a multi-threaded batch. The multi-threading is based on delinquency event ID and chunks for multi-threading are created based on numerical distribution of delinquency event ID. For more information about the batch, refer to *Oracle Revenue Management and Billing Batch Guide*.

Delinquency Process Maintenance

Once the delinquency process is created in the **Initiated** status, the system enables you to view the details of the delinquency process. Two new screens named **Delinquency Process Search** and **Delinquency Process** are introduced in this release. The **Delinquency Process Search** screen enables you to search for a delinquency process either using the details of the delinquency process or using the details of the account, person, or membership for which the delinquency process is created.

Note: The **Person ID** field while searching for a delinquency process is only applicable for the fully insured group business.

The **Delinquency Process** screen enables you to view the details of a delinquency process. It contains the following zones:

- **Delinquency Process Details** – Display the basic details of the delinquency process.
- **Additional Attributes** – Display the characteristics of the delinquency process.
- **Bills** – Lists the overdue bills of the account for which the delinquency process is created.
- **Events** - List all events that are included in the delinquency process.
- **Event Notifications** – Lists the notifications (i.e. To Do, customer contact), if any, created for the delinquency event.
- **Approval Details** – Lists the approval records that are created while performing manual actions at the delinquency process or delinquency event level.
- **Hold Details** – Lists the hold records that are created whenever you hold the delinquency process.
- **Log** – Lists the complete trail of actions performed on the delinquency process.

Manual Actions at the Delinquency Process Level

The **Delinquency Process** screen enables you to perform the following actions at the delinquency process level:

- **Edit** – Used when you want to add additional information for the delinquency process in the form of comments or characteristics.
- **Hold** – Used when you want to manually hold the delinquency process. The **Hold** button appears only when the delinquency process is in the **Initiated**, **Delinquency In Progress**, or **Pending Termination** status. While manually holding a delinquency process, you need to specify the following:
 - A hold end date which is later than the system date
 - The reason why you want to hold the delinquency process. Note that the list includes only those reasons which are defined for the **Initiate Hold** status of the **C1-DelinquencyProcess** business object.

The hold start date is set to the system date when you manually hold the delinquency process. On saving the changes, the status of the delinquency process is changed to the value (for example, **Initiate Hold**) specified in the **Delinquency Process Initiate Hold Status** attribute of the delinquency process preference. The system then checks whether the approval is required for the **Hold** action in the respective delinquency process type.

If the approval is not required while manually holding the delinquency process, the status of the delinquency process is immediately changed to the value (for example, **On Hold**) specified in the **Delinquency Process Hold Status** attribute of the delinquency process preference. A hold record is created for the delinquency process in the **Active** status. However, if the approval is required while manually holding the delinquency process, an approval record is created in the **Approval In Progress** status. A To Do is created using the approval To Do type to approve the manual hold and assigned to the users with the approval To Do role. In addition, the status of the delinquency process is changed to the value (for example, **Delinquency In Progress – Exception Raised**) specified in the **Delinquency Process Approval In Progress Status** attribute of the delinquency process preference. Note that the system continues to trigger the delinquency events on the trigger date even when the approval is in progress until the approver approves the manual hold.

If the approver approves the manual hold, the status of the delinquency process is changed to the value (for example, **On Hold**) specified in the **Delinquency Process Hold Status** attribute of the delinquency process preference. In addition, the system does the following:

- The hold start date is changed to the date when the approver approves the manual hold.
- The status of the approval record is changed to **Approved**.
- A hold record is created for the delinquency process in the **Active** status.

However, if the approver rejects the manual hold, the status of the approval record is changed to **Rejected** and the status of the delinquency process is changed to its previous status (i.e. **Initiated**, **Delinquency In Progress**, or **Pending Termination**).

- **Cancel** – Used when you want to manually cancel the delinquency process. The **Cancel** button appears only when the delinquency process is in the **Initiated**, **Delinquency In Progress**, or **On Hold** status. While manually canceling a delinquency process, you need to specify the cancel reason and comments, if any. You must specify a reason which is defined for the **Initiate Cancel** status of the **C1-DelinquencyProcess** business object in the **Status Reason** screen.

On manually canceling a delinquency process, the status of the delinquency process is set to the value (for example, **Initiate Cancel**) specified in the **Delinquency Process Initiate Cancel Status** attribute of the delinquency process preference. The system then checks whether the approval is required for the **Cancel** action in the respective delinquency process type. If the approval is not required while manually canceling the delinquency process, the status of the delinquency process is changed to the value (for example, **Canceled**) specified in the **Delinquency Process Canceled Status** attribute of the delinquency process preference.

However, if the approval is required while manually canceling the delinquency process, an approval record is created in the **Approval In Progress** status. A To Do is created using the approval To Do type to approve the manual cancellation and assigned to the users with the approval To Do role.

In addition, the status of the delinquency process is changed to the value (for example, **Delinquency In Progress – Exception Raised**) specified in the **Delinquency Process Approval In Progress Status** attribute of the delinquency process preference. Note that the system continues to trigger the delinquency events on the trigger date even when the approval is in progress until the approver approves the manual cancellation of the delinquency process.

If the approver approves the manual cancellation of the delinquency process, the status of the delinquency process is set to the value (for example, **Canceled**) specified in the **Delinquency Process Canceled Status** attribute of the delinquency process preference. In addition, the status of the approval record is changed to **Approved**. However, if the approver rejects the manual cancellation of the delinquency process, the status of the approval record is changed to **Rejected** and the status of the delinquency process is changed to its previous status (i.e. **Initiated**, **Delinquency In Progress**, or **On Hold**).

- **Release** – Used when you want to manually release the hold on the delinquency process. The **Release** button appears only when the delinquency process is in the **On Hold** status. While manually releasing the hold on the delinquency process, you are prompted to confirm the hold release. On confirming the hold release, the status of the delinquency process is changed to the value (for example, **Initiate Release**) specified in the **Delinquency Process Initiate Release Status** attribute of the delinquency process preference. The system then checks whether the approval is required for the **Release Hold** action in the respective delinquency process type. If the approval is not required while manually releasing the hold on the delinquency process, the status of the delinquency process is immediately changed to its previous status (i.e. **Initiated**, **Delinquency In Progress**, or **Pending Termination**). In addition, the hold end date is set to the date when you manually release the hold on the delinquency process. Finally, the status of hold record is changed to **Completed**.

However, if the approval is required while manually releasing the hold on the delinquency process, an approval record is created in the **Approval In Progress** status. A To Do is created using the approval To Do type to approve the manual hold release and assigned to the users with the approval To Do role. In addition, the status of the delinquency process is changed to the value (for example, **Delinquency In Progress – Exception Raised**) specified in the **Delinquency Process Approval In Progress Status** attribute of the delinquency process preference. If the approver approves the hold release, the status of the delinquency process is changed to its previous status (i.e. **Initiated**, **Delinquency In Progress**, or **Pending Termination**). The hold end date is set to the date when the approver approves the hold release. In addition, the status of the approval record is changed to **Approved** and the status of hold record is changed to **Completed**. However, if the approver rejects the hold release, the status of the approval record is changed to **Rejected** and the status of the delinquency process is changed to its previous status (i.e. **On Hold**).

- **Request Reinstatement** – This button is not applicable for the delinquency processes which are created for the fully insured individual business.
- **Edit Hold** – On clicking the **Edit** icon corresponding to the hold record of the delinquency process, you can edit the hold end date and hold reason. Note that the **Edit** icon appears only when following conditions are met:
 - The delinquency process is in the **On Hold** status.
 - The delinquency process is kept on hold manually and not automatically.

- The hold record is in the **Active** status.

While editing the hold details, you can specify the following:

- A hold end date which is later than the system date and which is later than the date which is the latest of the following:
 - Postpone credit review date
 - Latest schedule date specified in the payment agreement request
 - Latest schedule date specified among all the active promise to pay
- The reason why you want to hold the delinquency process. Note that the list includes only those reasons which are defined for the **Initiate Hold** status of the **C1-DelinquencyProcess** business object.

On saving the changes, the system checks whether the approval is required for the **Edit Hold** action in the respective delinquency process type. If the approval is not required while manually editing the hold details, the hold end date and hold reason of the delinquency process are updated immediately.

However, if the approval is required while manually editing the hold details, an approval record is created in the **Approval In Progress** status. A To Do is created using the approval To Do type to approve the changes and assigned to the users with the approval To Do role. The status of the delinquency process remains as **On Hold**. If the approver approves the changes, the hold end date and hold reason of the delinquency process are updated. In addition, the status of the approval record is changed to **Approved**. However, if the approver rejects the hold changes, the status of the approval record is changed to **Rejected** and the hold changes are not reflected in the delinquency process.

Points to Note:

The system considers the delinquency process preference which is specified in the **Delinquency Process Field Mapping** option type of the **DELINPROC** feature configuration. For more information about the feature configuration, refer to the [DELINPROC Feature Configuration](#) section.

If the required attributes are not defined in the delinquency process preference, erroneous results will occur.

The system considers the approval To Do type and approval To Do role specified in the respective delinquency process type. If the approval To Do role is not specified in the respective delinquency process type, the system considers the default To Do role associated with the approval To Do type.

Manual Actions at the Delinquency Event Level

You can also perform the following actions for each delinquency event of the delinquency process from the **Delinquency Process** screen:

- **Edit Trigger Date** – On clicking the **Edit Trigger Date** icon corresponding to a delinquency event, you can specify a new trigger date for the delinquency event. Note that the **Edit Trigger Date** icon appears only when the following conditions are met:

- The delinquency process is in the **Initiated** or **Delinquency In Progress** status
- The delinquency event trigger mode is set to **Manual** or **Automatic** in the respective delinquency process type
- The delinquency event is in the **Pending** status

Once you specify the new trigger date and save the changes, the system checks whether the approval is required for the **Edit Trigger Date** action in the respective delinquency process type. If the approval is not required while manually editing the trigger date, the trigger date of the delinquency event is updated immediately.

However, if the approval is required while manually editing the trigger date, an approval record is created for the delinquency event in the **Approval In Progress** status. A To Do is created using the approval To Do type to approve the changes and assigned to the users with the approval To Do role. Note that the icons in the **Trigger Events** and **Skip Event** columns (if any) corresponding to the delinquency event are removed immediately, so that users should not be able to perform any other actions on the delinquency event. The status of the delinquency process remains as **Initiated** or **Delinquency In Progress**. Note that the system continues to trigger the delinquency events on the trigger date even when the approval is in progress. If a delinquency event is triggered before the approval is received for the delinquency event, the status of the approval record is changed to **Expired**.

If the approver approves the changes, the trigger date of the delinquency event is updated. The icons in the **Trigger Events** and **Edit Trigger Date** columns (if any) corresponding to the delinquency event are displayed again. In addition, the status of the approval record is changed to **Approved**. However, if the approver rejects the changes, the status of the approval record is changed to **Rejected** and the trigger date of the delinquency event is not updated. However, the icons in the **Trigger Events** and **Edit Trigger Date** columns (if any) corresponding to the delinquency event are displayed again.

- **Skip Event** – On clicking the **Skip Event** icon corresponding to a delinquency event, the system checks whether the approval is required for the **Skip Event** action in the respective delinquency process type. Note that the **Skip Event** icon appears only when the following conditions are met:
 - The delinquency process is in the **Initiated** or **Delinquency In Progress** status
 - The delinquency event trigger mode is set to **Manual** or **Manual or Automatic** in the respective delinquency process type
 - The delinquency event is in the **Pending** status

If the approval is not required while manually skipping the delinquency events, the status of the delinquency event is changed to **Skipped** immediately. In addition, the icons in the **Trigger Events** and **Edit Trigger Date** columns (if any) corresponding to the delinquency event are also removed.

However, if the approval is required while manually skipping the delinquency events, an approval record is created for the delinquency event in the **Approval In Progress** status. A To Do is created using the approval To Do type to approve the changes and assigned to the users with the approval To Do role.

Note that the icons in the **Trigger Events** and **Edit Trigger Date** columns (if any) corresponding to the delinquency event are removed immediately, so that users should not be able to perform any other actions on the delinquency event. The status of the delinquency process remains as **Initiated** or **Delinquency In Progress**. Note that the system continues to trigger the delinquency events on the trigger date even when the approval is in progress. If a delinquency event is triggered before the approval is received for the delinquency event, the status of the approval record is changed to **Expired**.

If the approver approves the skip action, the status of the delinquency event is changed to **Skipped** immediately. In addition, the status of the approval record is changed to **Approved**. Note that the icons in the **Trigger Events** and **Edit Trigger Date** columns corresponding to the delinquency event are not displayed again. However, if the approver rejects the skip action, the status of the approval record is changed to **Rejected**. The status of the delinquency event remains as **Pending**. In addition, the icons in the **Trigger Events** and **Edit Trigger Date** columns (if any) corresponding to the delinquency event are displayed again.

- **Trigger Event** - On clicking the **Trigger Event** icon corresponding to a delinquency event, the system checks whether the approval is required for the **Trigger Event** action in the respective delinquency process type. Note that the **Trigger Event** icon appears only when the following conditions are met:
 - The delinquency process is in the **Initiated** or **Delinquency In Progress** status
 - The delinquency event trigger mode is set to **Manual** or **Manual or Automatic** in the respective delinquency process type
 - The delinquency event is in the **Pending** status

If the approval is not required while manually triggering the delinquency events, the algorithms attached to the **Event Activation** system event of the respective delinquency event type are executed immediately in the specified order. The status of the delinquency event is changed to **Completed** when all algorithms attached to the **Event Activation** system event are successfully executed. In addition, the icons in the **Skip Event** and **Edit Trigger Date** columns (if any) corresponding to the delinquency event are also removed.

However, if the approval is required while manually triggering the delinquency events, an approval record is created for the delinquency event in the **Approval In Progress** status. A To Do is created using the approval To Do type to approve the changes and assigned to the users with the approval To Do role. Note that the icons in the **Skip Event** and **Edit Trigger Date** columns (if any) corresponding to the delinquency event are removed immediately, so that users should not be able to perform any other actions on the delinquency event.

The status of the delinquency process remains as **Initiated** or **Delinquency In Progress**. Note that the system continues to trigger the delinquency events on the trigger date even when the approval is in progress. If a delinquency event is triggered before the approval is received for the delinquency event, the status of the approval record is changed to **Expired**.

If the approver approves the manual trigger action, the algorithms attached to the **Event Activation** system event of the respective delinquency event type are executed immediately in the specified order.

The status of the delinquency event is changed to **Completed** when all algorithms attached to the **Event Activation** system event are successfully executed. In addition, the status of the approval record is changed to **Approved**. Note that the icons in the **Skip Event** and **Edit Trigger Date** columns corresponding to the delinquency event are not displayed again. However, if the approver rejects the manual trigger action, the status of the approval record is changed to **Rejected**. The status of the delinquency event remains as **Pending**. In addition, the icons in the **Skip Event** and **Edit Trigger Date** columns (if any) corresponding to the delinquency event are displayed again.

Note that the algorithms attached to the **Monitor Completed Event** system event of the respective delinquency event type are not executed in the online mode when the status of the delinquency event is changed to **Completed**. These algorithms are executed when you invoke the **C1-DPMON** batch.

Whenever you manually execute the subsequent events (non-first event) in the delinquency process, the system first checks whether the algorithms attached to the **Monitor Completed Event** system event of the previous delinquency event is successfully executed or not. If an algorithm attached to the **Monitor Completed Event** system event of the previous delinquency event is not executed successfully, the system displays a warning message that indicates the previous event's monitoring algorithms are not executed successfully and confirms whether you want to continue in such scenario. If you click **Yes**, the system ignores the previous event's monitoring algorithm and executes the algorithms attached to the **Event Activation** system event of the current delinquency event (which is triggered). However, if you click **No**, the system does not execute the algorithms attached to the **Event Activation** system event of the current delinquency event. Note that if there are no algorithms attached to the **Monitor Completed Event** system event of the previous delinquency event, the system will directly execute the algorithms attached to the **Event Activation** system event of the current delinquency event.

Note: The system considers the approval To Do type and approval To Do role specified in the respective delinquency process type. If the approval To Do role is not specified in the respective delinquency process type, the system considers the default To Do role associated with the approval To Do type.

Delinquency Process Approval

At a time, the system creates an approval record for only one manual action at the delinquency process or delinquency event level. Therefore, if an approval record exists for a delinquency process, the system does not allow you to perform other manual action on the delinquency process until the respective approval record's status changes to **Approved** or **Rejected**. Similarly, if an approval record exists for a delinquency event, the system does not allow you to perform other manual action on the delinquency event until the respective approval record's status changes to **Approved** or **Rejected**. However, you can still perform a manual action for other delinquency events of the delinquency process.

The system enables you to withdraw an approval request for any manual action at the delinquency process or delinquency event level. Only the users of the user group to which the submitter belongs can withdraw the approval request which is in the **Approval In Progress** status. On confirming the withdrawal of an approval request, the status of the approval record is changed to **Withdrawn**.

The system enables you to perform the following actions for an approval record from the **Delinquency Process** screen:

- **Add Comments** – Used when you want to add additional information about the approval record. Both the submitter and approver can add the comments.
- **View Details** – Used to view the details of the manual action which is performed by the submitter. If the approval record is created while manually editing the trigger date of a delinquency event, the system enables you to view the old and new trigger date of the delinquency event. If the approval record is created while manually holding a delinquency process, the system enables you to view the hold end date and hold reason. However, if the approval record is created while manually editing the hold details of the delinquency process, the system enables you to view the old and new hold end dates and hold reasons. Both the submitter and approver can view the details of an approval record.
- **Approve** – Used when you want to approve the manual action performed at the delinquency process or delinquency event level. Only users with the approver To Do role can approve the approval request.
- **Reject** - Used when you want to reject the manual action performed at the delinquency process or delinquency event level. Only users with the approver To Do role can reject the approval request.
- **Withdraw** – Used when you want to withdraw the approval request. Only the users of the user group to which the submitter belongs can withdraw the approval request which is in the **Approval In Progress** status.

Points to Note:

If the delinquency process is in the **On Hold** status, you cannot approve or reject any other manual actions except the **Release Hold** and **Edit Hold** actions. All other manual actions can be reviewed and accordingly approved or rejected by the approver only after the hold on the delinquency process is released.

When a delinquency process is canceled, the status of the pending approval records is changed to **Expired**.

Canceling a Delinquency Process

Oracle Revenue Management and Billing enables you to cancel a delinquency process of a person or an account manually or automatically. You can manually cancel a delinquency process whenever required. In addition, the system automatically cancels a delinquency process when either of the following condition is met:

- Payments or adjustments are made against the overdue bills, for which the delinquency process is created, and as a result the unpaid balance of the overdue bills is within the tolerance limit
- Collection class of the account, for which the delinquency process is created, is changed
- Voluntary termination is received for an individual membership (which is billed to the delinquent account) with a termination date that is earlier than or equal to the calculated termination date
- Voluntary cancellation is received for an individual membership (which is billed to the delinquent account)

To enable the automatic cancellation process for the delinquency processes which are created using a particular delinquency process type, you need to attach the **C1-CANDELPRC** algorithm to the **Cancel Criteria** system event of the respective delinquency process type. This algorithm is invoked when you execute the **C1-DPMON** batch. For more information about the automatic cancellation process, refer to the respective algorithm description in the [Delinquency Process Type Algorithms](#) section.

The **Cancel** button in the **Delinquency Process** screen enables you to manually cancel a delinquency process. It appears only when the delinquency process is in the **Initiated**, **Delinquency In Progress**, or **On Hold** status. On manually canceling a delinquency process, the status of the delinquency process is changed to the value (for example, **Initial Cancel**) specified in the **Delinquency Process Initiate Cancel Status** attribute of the delinquency process preference. For more information about the manual cancellation process, refer to the [Manual Actions at the Delinquency Process Level](#) section.

Once a delinquency process is manually or automatically canceled, the status of the delinquency process is changed to the value (for example, **Canceled**) specified in the **Delinquency Process Canceled Status** attribute of the delinquency process preference. Note that a delinquency process in the **Pending Termination** status cannot be canceled manually but only automatically through the **C1-DPMON** batch.

Note: The system considers the delinquency process preference which is specified in the **Delinquency Process Field Mapping** option type of the **DELINPROC** feature configuration. For more information about the feature configuration, refer to the [DELINPROC Feature Configuration](#) section.

Once the status of the delinquency process is changed to **Canceled**, the system then executes the **C1-DPCAN** algorithm attached to the **Canceled** status in the lifecycle of the **C1-DelinquencyProcess** business object. The **C1-DPCAN** algorithm does the following:

- Creates a To Do notification with the cancel reason
- If there is any hold record for the delinquency process in the **Active** status, this algorithm changes the status of the hold record to **Complete**.
- If there is any pending record for approval at the delinquency event or process level, this algorithm changes the status of pending approval records to **Expired**.

This algorithm contains the following parameters:

- **To Do Type** – Used to indicate the To Do type using which you want to create a To Do when the delinquency process is canceled. This parameter is required.
- **To Do Role** – Used to indicate that users with the specified To Do role must receive the To Do notification. This parameter is optional. If you do not specify this parameter, this algorithm considers the default To Do role associated with the To Do type.
- **To Do Entry required for Individual Delinquency (Valid Values - Y/N)** – Used to indicate whether the To Do notification should be created on canceling a delinquency process for the fully insured individual business. The valid values are **Y** and **N**. Note that this parameter is not applicable for the fully insured group business. This algorithm will create a To Do notification on canceling a delinquency process for the fully insured group business irrespective of whether this parameter is set to **Y** or **N**.

Resuming a Delinquency Process

Oracle Revenue Management and Billing enables you to resume a delinquency process which was canceled due to a payment or adjustment made against the overdue bill for which the delinquency process was created. The system enables you to resume a delinquency process only when the respective payment or adjustment is canceled due to non-sufficient funds. The insurance company may come across the following scenarios:

- Scenarios wherein the delinquency process is canceled in between events when the status of the delinquency process is **Initiated**, **Delinquency In Progress**, or **On Hold**
- Scenarios wherein the delinquency process is canceled after the delinquency process moves to the **Customer Reinstated** status due the individual membership reinstatement

The system enables you to handle both these scenarios when a payment or adjustment made against a bill is canceled.

Resuming a Delinquency Process on Payment Cancellation

To automatically resume a delinquency process when a payment made against the respective overdue bill is canceled, you need to attach the following algorithms to the **Payment Cancellation** system event of the required customer classes:

- **C1-CDPPAYCAN** - This algorithm is invoked when a payment is canceled. It checks whether the **Delinquency Process (C1-DELPO)** characteristic is defined on the payment. If the **C1-DELPO** characteristic is not defined on the payment, the system creates a customer contact for the main customer of the account for which the payment was made. In addition, the system stamps the payment ID as a characteristic using the given characteristic type on the customer contact and the customer contact ID as a characteristic using the given characteristic type on the payment.

However, if the **C1-DELPO** characteristic is defined on the payment, the system derives the delinquency process ID from characteristic value. It then validates whether the delinquency process exist with the given ID in the system. Note that if the delinquency process does not exist, no further action takes place on the payment cancellation. However, if the delinquency process ID is successfully validated, the system checks the following:

- The status of the delinquency process is set to the value (for example, **Canceled**) specified in the **Delinquency Process Canceled Status** attribute of the delinquency process preference.
- The related delinquency process does not exist for the delinquency process.

Note: The system considers the delinquency process preference which is specified in the **Delinquency Process Field Mapping** option type of the **DELINPROC** feature configuration. For more information about the feature configuration, refer to the [DELINPROC Feature Configuration](#) section.

If either or both the conditions are not satisfied, no further action takes place on the payment cancellation. However, if both the conditions are satisfied, the system invokes the **C1-CANDELPRC** algorithm attached to the **Cancel Criteria** system event of the respective delinquency process type.

If the cancel criteria is met, the delinquency process remains in the existing status and no further action takes place on the payment cancellation. However, if the cancel criteria is not met, the status of the delinquency process is set to the status prior to cancellation (i.e. **Initiated**, **Delinquency In Progress**, or **On Hold**). If the status of the delinquency process before cancellation was **On Hold**, then the status of the hold record is changed to **Active**. The system also creates a customer contact for the main customer of the account for which the payment was made. In addition, the system does the following:

- Adds a log entry for the customer contact in the delinquency process
- Stamps the customer contact ID as a characteristic using the given characteristic type on the payment
- Stamps the payment ID as a characteristic using the given characteristic type on the customer contact
- Stamps the delinquency process ID using the **C1-DELPO** characteristic type on the customer contact

This algorithm derives the preferred contact method for the customer contact. While deriving the preferred contact method, the algorithm first fetches the main customer's bill route type on the account and then fetches the bill routing method of the bill route type. Once the bill routing method is derived, the system considers the preference which is specified in the **Routing Method – Contact Method Mapping** option type of the **DELINPROC** feature configuration. For more information about the feature configuration, refer to the [DELINPROC Feature Configuration](#) section. It then checks whether any preferred contact method is mapped to the bill routing method in the preference.

If the preferred contact method is mapped to the bill routing method in the preference, the system stamps it on the customer contact. However, if the preferred contact method is not mapped to the bill routing method in the preference, the system considers the default contact method as the preferred contact method and stamps it on the customer contact.

This algorithm contains the following parameters:

- **Customer Contact Type** - Used to indicate the customer contact type using which you want to create the customer contact.
- **Customer Contact Class** - Used to indicate the customer contact class to which the customer contact type belongs.
- **Default Contact Method** - Used to indicate the contact method that you want to use when the preferred contact method cannot be derived for the routing method from the preference.
- **Customer Contact Characteristic Type** - Used to indicate the characteristic type using which you want to store the customer contact ID on the payment. Here, you must specify a characteristic type where the characteristic entity is set to **Payment**.
- **Payment Characteristic For Customer Contact** - Used to indicate the characteristic type using which you want to store the payment ID on the customer contact. Here, you must specify a characteristic type where the characteristic entity is set to **Customer Contact**.

All the above parameters are mandatory.

- **C1-RODPDUEPX** - This algorithm is invoked when a payment is canceled. It derives the reason why the payment was canceled and validates whether it is listed as a value in the **Excluded Payment Cancel Reason List** parameter. If the payment cancel reason is listed in the **Excluded Payment Cancel Reason List** parameter, no further action takes place on the payment cancellation. However, if the payment cancel reason is not listed in the **Excluded Payment Cancel Reason List** parameter, the system checks whether the **Delinquency Process (C1-DELPO)** characteristic is defined on the payment. If the **C1-DELPO** characteristic is not defined on the payment, no further action takes place on the payment cancellation.

However, if the **C1-DELPO** characteristic is defined on the payment, the system derives the delinquency process ID from characteristic value. It then validates whether the delinquency process exist with the given ID in the system. If the delinquency process does not exist, no further action takes place on the payment cancellation. However, if the delinquency process ID is successfully validated, the system checks whether the status of the delinquency process is set to the value (for example, **Customer Reinstated**) specified in the **Delinquency Process Customer Reinstated Status** attribute of the delinquency process preference. If so, the system checks the value of the **Delinquency Process Resume Switch (Valid Values – Y,N)** parameter. If the **Delinquency Process Resume Switch (Valid Values – Y,N)** parameter is set to **Y**, the system does the following:

- Sets the status of the delinquency process to the value (for example, **Reinstatement Payment/Adjustment Canceled**) specified in the **Delinquency Reinstatement Payment/Adjustment Canceled Status** attribute of the delinquency process preference

However, if the **Delinquency Process Resume Switch (Valid Values – Y,N)** parameter is set to **N**, the system does the following:

- If a delinquency process created using the given delinquency process type already exists for the account, the system adds the overdue bill for which the payment is canceled in the existing delinquency process. In addition, a log entry is created for the overdue bill which is added in the existing delinquency process. Note that the system considers an existing delinquency process which is in the status that is listed in the **Delinquency Process Status For Existing Delinquency Process** parameter of the **C1-DLNQMNRL** algorithm attached to the delinquency control of the person's or account's collection class, respectively.
- If a delinquency process does not exist for the account, the system creates a new delinquency process using the given delinquency process type. It adds the overdue bill for which the payment is canceled in the new delinquency process. Also, if there are any other overdue bills of the account that do not meet the tolerance limit are also added in the delinquency process. The status of the delinquency process is set to the value (for example, **Initiated**) specified in the **Delinquency Process Initiated Status** attribute of the delinquency process preference. In addition, a log entry is created for each overdue bill which is added in the delinquency process.

Note: The system considers the delinquency process preference which is specified in the **Delinquency Process Field Mapping** option type of the **DELINPROC** feature configuration. For more information about the feature configuration, refer to the [DELINPROC Feature Configuration](#) section.

This algorithm contains the following parameters:

- **Excluded Payment Cancel Reason List** – Used to specify a list of comma-separated payment cancel reasons. The system will then skip the payments which are canceled using the given payment cancel reason and will not take any actions on the payment cancellation. You can specify maximum five comma-separated values for this parameter. Here, you must specify a payment cancel reason which is already defined in the system.
- **Delinquency Process Resume Switch (Valid Values – Y,N)** – Used to indicate whether the original delinquency process which was canceled due to the payment should be resumed when the payment is canceled. The valid values are – **Y** and **N**. This parameter is mandatory.
- **Delinquency Process Type** – Used to indicate the delinquency process type using which a new process delinquency process should be created for the account when the payment is canceled. It is also used to check whether any existing delinquency process created using the given delinquency process type already exists in the system. Here, you must specify a delinquency process type which is already defined in the system. This parameter is required only when the **Delinquency Process Resume Switch (Valid Values – Y,N)** parameter is set to **N**.

Resuming a Delinquency Process on Adjustment Cancellation

To automatically resume a delinquency process when an adjustment made against the respective overdue bill is canceled, you need to attach the following algorithms to the **Adjustment Cancellation** system event of the required adjustment types:

- **C1-DPADJCAN** - This algorithm is invoked when an adjustment is canceled. It checks whether the **Delinquency Process (C1-DELPO)** characteristic is defined on the adjustment. If the **C1-DELPO** characteristic is not defined on the adjustment, the system creates a customer contact for the main customer of the account for which the adjustment was created. In addition, the system stamps the adjustment ID as a characteristic using the given characteristic type on the customer contact and the customer contact ID as a characteristic using the given characteristic type on the adjustment.

However, if the **C1-DELPO** characteristic is defined on the adjustment, the system derives the delinquency process ID from characteristic value. It then validates whether the delinquency process exist with the given ID in the system. Note that if the delinquency process does not exist, no further action takes place on the adjustment cancellation. However, if the delinquency process ID is successfully validated, the system checks the following:

- The status of the delinquency process is set to the value (for example, **Canceled**) specified in the **Delinquency Process Canceled Status** attribute of the delinquency process preference.
- The related delinquency process does not exist for the delinquency process.

Note: The system considers the delinquency process preference which is specified in the **Delinquency Process Field Mapping** option type of the **DELINPROC** feature configuration. For more information about the feature configuration, refer to the [DELINPROC Feature Configuration](#) section.

If either or both the conditions are not satisfied, no further action takes place on the adjustment cancellation. However, if both the conditions are satisfied, the system invokes the **C1-CANDELPRC** algorithm attached to the **Cancel Criteria** system event of the respective delinquency process type.

If the cancel criteria is met, the delinquency process remains in the existing status and no further action takes place on the adjustment cancellation. However, if the cancel criteria is not met, the status of the delinquency process is set to the status prior to cancellation (i.e. **Initiated**, **Delinquency In Progress**, or **On Hold**). If the status of the delinquency process before cancellation was **On Hold**, then the status of the hold record is changed to **Active**. The system also creates a customer contact for the main customer of the account for which the adjustment was created. In addition, the system does the following:

- Adds a log entry for the customer contact in the delinquency process
- Stamps the customer contact ID as a characteristic using the given characteristic type on the adjustment
- Stamps the adjustment ID as a characteristic using the given characteristic type on the customer contact
- Stamps the delinquency process ID using the **C1-DELPO** characteristic type on the customer contact

This algorithm derives the preferred contact method for the customer contact. While deriving the preferred contact method, the algorithm first fetches the main customer's bill route type on the account and then fetches the bill routing method of the bill route type. Once the bill routing method is derived, the system considers the preference which is specified in the **Routing Method – Contact Method Mapping** option type of the **DELINPROC** feature configuration. For more information about the feature configuration, refer to the [DELINPROC Feature Configuration](#) section. It then checks whether any preferred contact method is mapped to the bill routing method in the preference.

If the preferred contact method is mapped to the bill routing method in the preference, the system stamps it on the customer contact. However, if the preferred contact method is not mapped to the bill routing method in the preference, the system considers the default contact method as the preferred contact method and stamps it on the customer contact.

This algorithm contains the following parameters:

- **Customer Contact Type** - Used to indicate the customer contact type using which you want to create the customer contact.
- **Customer Contact Class** - Used to indicate the customer contact class to which the customer contact type belongs.
- **Default Contact Method** - Used to indicate the contact method that you want to use when the preferred contact method cannot be derived for the routing method from the preference.
- **Customer Contact Characteristic Type** - Used to indicate the characteristic type using which you want to store the customer contact ID on the adjustment. Here, you must specify a characteristic type where the characteristic entity is set to **Adjustment**.
- **Adjustment Characteristic For Customer Contact** - Used to indicate the characteristic type using which you want to store the adjustment ID on the customer contact. Here, you must specify a characteristic type where the characteristic entity is set to **Customer Contact**.

All the above parameters are mandatory.

- **C1-RODPONADX** - This algorithm is invoked when an adjustment is canceled. It derives the reason why the adjustment was canceled and validates whether it is listed as a value in the **Excluded Adjustment Cancel Reason List** parameter. If the adjustment cancel reason is listed in the **Excluded Adjustment Cancel Reason List** parameter, no further action takes place on the adjustment cancellation. However, if the adjustment cancel reason is not listed in the **Excluded Adjustment Cancel Reason List** parameter, the system checks whether the **Delinquency Process (C1-DELPO)** characteristic is defined on the adjustment. If the **C1-DELPO** characteristic is not defined on the adjustment, no further action takes place on the adjustment cancellation.

However, if the **C1-DELPO** characteristic is defined on the adjustment, the system derives the delinquency process ID from characteristic value. It then validates whether the delinquency process exist with the given ID in the system. If the delinquency process does not exist, no further action takes place on the adjustment cancellation. However, if the delinquency process ID is successfully validated, the system checks whether the status of the delinquency process is set to the value (for example, **Customer Reinstated**) specified in the **Delinquency Process Customer Reinstated Status** attribute of the delinquency process preference. If so, the system checks the value of the **Delinquency Process Resume Switch** parameter. If the **Delinquency Process Resume Switch** parameter is set to **Y**, the system does the following:

- Sets the status of the delinquency process to the value (for example, **Reinstatement Payment/Adjustment Canceled**) specified in the **Delinquency Reinstatement Payment/Adjustment Canceled Status** attribute of the delinquency process preference

However, if the **Delinquency Process Resume Switch** parameter is set to **N**, the system does the following:

- If a delinquency process created using the given delinquency process type already exists for the account, the system adds the overdue bill for which the adjustment is canceled in the existing delinquency process. In addition, a log entry is created for the overdue bill which is added in the existing delinquency process. Note that the system considers an existing delinquency process which is in the status that is listed in the **Delinquency Process Status For Existing Delinquency Process** parameter of the **C1-DLNQMNRL** algorithm attached to the delinquency control of the person's or account's collection class.
- If a delinquency process does not exist for the account, the system creates a new delinquency process using the given delinquency process type. It adds the overdue bill for which the adjustment is canceled in the new delinquency process. Also, if there are any other overdue bills of the account that do not meet the tolerance limit are also added in the delinquency process. The status of the delinquency process is set to the value (for example, **Initiated**) specified in the **Delinquency Process Initiated Status** attribute of the delinquency process preference. In addition, a log entry is created for each overdue bill which is added in the delinquency process.

Note: The system considers the delinquency process preference which is specified in the **Delinquency Process Field Mapping** option type of the **DELINPROC** feature configuration. For more information about the feature configuration, refer to the [DELINPROC Feature Configuration](#) section.

This algorithm contains the following parameters:

- **Excluded Adjustment Cancel Reason List** – Used to specify a list of comma-separated adjustment cancel reasons. The system will then skip the adjustments which are canceled using the given adjustment cancel reason and will not take any actions on the adjustment cancellation. You can specify maximum five comma-separated values for this parameter. Here, you must specify an adjustment cancel reason which is already defined in the system.
- **Delinquency Process Resume Switch** – Used to indicate whether the original delinquency process which was canceled due to the adjustment should be resumed when the adjustment is canceled. The valid values are – **Y** and **N**. This parameter is mandatory.
- **Delinquency Process Type** – Used to indicate the delinquency process type using which a new process delinquency process should be created for the account when the adjustment is canceled. It is also used to check whether any existing delinquency process created using the given delinquency process type already exists in the system. Here, you must specify a delinquency process type which is already defined in the system. This parameter is required only when the **Delinquency Process Resume Switch** parameter is set to **N**.

Recalculating Trigger Dates for Pending Events

Whenever a delinquency process moves from the **Canceled** status to **Initiated**, **Delinquency In Progress**, or **On Hold** on the payment or adjustment cancellation, the system executes the **C1-RECALCTG** algorithm attached to the **Canceled** status of the **C1-DelinquencyProcess** business object. This algorithm recalculates the trigger date for the delinquency events which are in the **Pending** status. It derives the difference between the delinquency process resume date and delinquency process cancellation date. These extra days are then added to the existing trigger date of the pending delinquency events and thereby the new trigger dates are calculated for the remaining events. If the **Consider Work Days** option is selected in the respective delinquency process type, the system derives the work calendar of the division to which the account belongs. The system then checks whether the new trigger date falls on the working (i.e. business) day. If the new trigger date does not fall on the working (i.e. business) day, the system will accordingly add days to the new trigger date to ensure that it falls on the next working day.

Membership Cancellation Outbound Message

The **C1-MemberCancRequest** business object is newly introduced in this release. It enables you to create a cancellation outbound message for an individual membership. Note that this business object is only applicable for the fully insured individual business.

To create the cancellation outbound messages for individual memberships through the **Delinquency Management** feature, you need to do the following:

- Create an external system to which you want to send the outbound messages. Note that the implementation team needs to design how the outbound message should be sent to the external system based on the customer requirements (i.e. through a batch control or a real-time service). Also, any response from the external system on receiving the outbound message should be captured in ORMB.
- Add the external system in the **C1-SourceSystemLookup** extendable lookup

- Create an outbound message type using which you want to create a cancellation outbound message for an individual membership. Here, you must specify the following:
 - The **C1-MemberCancRequest** business object using which you want to create the cancellation outbound messages for the individual memberships.
 - An external system to which you want to send the cancellation outbound messages that are created using the outbound message type.
- Create a delinquency process type using which you want to create a delinquency process to initiate cancellation request through an outbound message. For more information on how to design a delinquency process type to initiate cancellation request for an individual membership, refer to the [Membership Cancellation Delinquency Process](#) section.

Once the status reason of the individual memberships is set to the status reason which is specified in the **Awaiting Membership Cancellation Reason** attribute of the delinquency process preference, you need to execute the **C1-MTOMT** batch to generate the cancellation outbound messages for the individual memberships. For more information about the batch, refer to the [Outbound Message Generation for Delinquency Management](#) section.

Note: The system considers the delinquency process preference which is specified in the **Delinquency Process Field Mapping** option type of the **DELINPROC** feature configuration. For more information about the feature configuration, refer to the [DELINPROC Feature Configuration](#) section.

Membership Termination Outbound Message

The **C1-MemberTermRequest** business object is newly introduced in this release. It enables you to create a termination outbound message for an individual membership. Note that this business object is only applicable for the fully insured individual business.

To create the termination outbound messages for individual memberships through the **Delinquency Management** feature, you need to do the following:

- Create an external system to which you want to send the outbound messages. Note that the implementation team needs to design how the outbound message should be sent to the external system based on the customer requirements (i.e. through a batch control or a real-time service). Also, any response from the external system on receiving the outbound message should be captured in ORMB.
- Add the external system in the **C1-SourceSystemLookup** extendable lookup
- Create an outbound message type using which you want to create a termination outbound message for an individual membership. Here, you must specify the following:
 - The **C1-MemberTermRequest** business object using which you want to create the termination outbound messages for the individual memberships.
 - An external system to which you want to send the termination outbound messages that are created using the outbound message type.

- Create a delinquency process type using which you want to create a delinquency process to initiate termination request through an outbound message. For more information on how to design a delinquency process type to initiate termination request for an individual membership, refer to the [Membership Termination Delinquency Process](#) section.

Once the status reason of the individual memberships is set to the status reason which is specified in the **Membership Termination Reason for Delinquency** attribute of the delinquency process preference, you need to execute the **C1-MTOMT** batch to generate the termination outbound messages for the individual memberships. For more information about the batch, refer to the [Outbound Message Generation for Delinquency Management](#) section.

Note: The system considers the delinquency process preference which is specified in the **Delinquency Process Field Mapping** option type of the **DELINPROC** feature configuration. For more information about the feature configuration, refer to the [DELINPROC Feature Configuration](#) section.

Creating Customer Contacts for Group or Individual Memberships

If you send a debt collection letter to a delinquent customer or account through a delinquency process, you can configure the system to create a customer contact for the main subscriber of each membership which is billed to the delinquent customer or account. The system creates a customer contact for the main subscriber of each membership whenever an event is triggered in the delinquency process to send a debt collection letter.

Two fields named **Send Membership Level Notification Source** and **Send Membership Level Notification** field are available when you add a delinquency event type in a delinquency process type. You can set the **Send Membership Level Notification Source** field to either **Delinquency Process Type** or **Algorithm**. If you set the **Send Membership Level Notification Source** field to **Delinquency Process Type**, the system derives the value for the **Send Membership Level Notification** field corresponding to the respective event type from the delinquency process type. However, if you set the **Send Membership Level Notification Source** field to **Algorithm**, the system derives the value for the **Send Membership Level Notification** field from a business rule. To derive the value from the business rule, the system uses the the **C1-DERMLNBR** algorithm attached to the **Send Member Level Notification Option Derivation** system event in the respective delinquency process type. For more information about the **C1-DERMLNBR** algorithm, refer to the [Delinquency Process Type Algorithms](#) section. Note that these fields are applicable only when the delinquency event is triggered in the delinquency process to send a debt collection letter (such as, warning, dunning, or harsh letters) to the delinquent customer or account.

If the value of the **Send Membership Level Notification** field is set to **Send Membership Level Notification Required** either in the delinquency process type or in the business rule depending on the send membership level notification source, the system derives the memberships billed to the delinquent customer or account. It then checks whether the number of memberships derived is less than or equal to the value specified in the **Member Level Notification Deferred Processing Threshold** parameter of the **C1-DLQSENDCC** algorithm. If the number of memberships derived is less than or equal to the specified value, the system creates a customer contact for the main subscriber of each derived membership in the real time (i.e. immediately). In addition, the status of the delinquency event is changed to **Complete**.

However, if no value is defined for the **Member Level Notification Deferred Processing Threshold** parameter or if the number of memberships derived is greater than the specified value, the system changes the status of the delinquency event to **Pending Contact Creation**.

A new batch named **C1-MBRNT** is introduced in this release. This batch considers the delinquency processes which have at least one delinquency event in the **Pending Contact Creation** status. It then derives the memberships billed to the delinquent customer or account and creates a customer contact for the main subscriber of each derived membership.

In addition, this algorithm does the following:

- Creates a log entry for each customer contact in the delinquency process.
- Stamps the delinquency process ID as a characteristic using the given characteristic type on the customer contact.
- Stamps the membership ID as a characteristic using the given characteristic type on the customer contact.

Note: Once a customer contact is created, a record is added in the **C1_DELIN_PROC_EVT_NOTIF** table where the **DP_EVT_NOTIF_TYPE_FLG** column is set to CC and the **DP_EVT_NOTIF_ID** column is set to the customer contact ID.

This batch derives the preferred contact method for the customer contact. While deriving the preferred contact method, the batch first fetches the main customer's bill route type on the account and then fetches the bill routing method of the bill route type. Once the bill routing method is derived, the system considers the preference which is specified in the **Routing Method – Contact Method Mapping** option type of the **DELINPROC** feature configuration. For more information about the feature configuration, refer to the [DELINPROC Feature Configuration](#) section.

It then checks whether any preferred contact method is mapped to the bill routing method in the preference. If the preferred contact method is mapped to the bill routing method in the preference, the system stamps it on the customer contact. However, if the preferred contact method is not mapped to the bill routing method in the preference, the system considers the default contact method as the preferred contact method and stamps it on the customer contact.

This batch is a multi-threaded batch. The multi-threading is based on membership ID and chunks for multi-threading are created based on numerical distribution of membership ID. This batch contains the following parameters:

Parameter Name	Mandatory (Yes or No)	Description
Chunk Size	No	Used to specify the number of memberships you want to process in each work unit.
Customer Contact Type	Yes	Used to indicate the customer contact type using which you want to create the customer contact.

Parameter Name	Mandatory (Yes or No)	Description
		<p>Note: You must specify a customer contact type with the appropriate letter template depending on the type of letter you want to send to the customer.</p>
Customer Contact Class	Yes	<p>Used to indicate the customer contact class to which the customer contact type belongs.</p> <p>Note: You must specify a customer contact class which is already defined in the system.</p>
Default Contact Method	Yes	Used to indicate the contact method that you want to use when the preferred contact method cannot be derived for the routing method from the preference.
Delinquency Process Type	No	<p>Used when you want to create customer contacts for those delinquency processes which are created using a particular delinquency process type.</p> <p>Note: You must specify a delinquency process type which is already defined in the system.</p>
Membership Characteristic Type	Yes	<p>Used to indicate the characteristic type using which you want to store the membership ID on the customer contact.</p> <p>Note: You must specify a characteristic type where the characteristic entity is set to Customer Contact.</p>
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

DELINPROC Feature Configuration

A new feature configuration named **DELINPROC** is introduced for the **Delinquency Management** feature. It contains the following option types:

- **Delinquency Process Field Mapping** – Used to specify a delinquency process preference which is created using the **Delinquency Process** preference category through the **Field Mapping** screen. The system then avails the attributes from the preference at various stages in the lifecycle of a delinquency process. This option type is applicable for both the fully insured group and the fully insured individual businesses. For more information about the preference, refer to the [Delinquency Process Preference](#) section.
- **Policy Characteristic Type for Customer Contact** - Used to indicate the characteristic type using which you want to store the policy ID on the customer contact. Here, you must specify a characteristic type where the characteristic entity is set to **Customer Contact**. This option type is referred in the **C1-NOTADREC** and **C1-NOTADCAN** algorithms attached to a customer class. It is required when the group customer wants to allocate advance deposit to settle its overdue bills through a delinquency process. It is only applicable for the fully insured group business.
- **Routing Method – Contact Method Mapping** – Used to specify a routing method – contact method mapping preference which is created using the **C1_BillRouteCntMethMap** business object through the **Field Mapping** screen. The system then derives the preferred contact method for the bill routing method from the routing method – contact method mapping preference while creating a customer contact for a person or an account during the delinquency management. This option type is applicable for both the fully insured group and the fully insured individual businesses. For more information about the preference, refer to the [Routing Method – Contact Method Mapping Preference](#) section.

Outbound Message Generation for Delinquency Management

A new batch named **C1-MTOMT** is introduced in this release. This batch is used to create the termination and cancellation outbound messages whenever the termination or cancellation process is initiated for the individual memberships through the **Delinquency Management** feature. It creates the termination and cancellation outbound messages using the given outbound message types for the individual memberships which have the given status and status reasons. In addition, it adds a log for the outbound message in the respective individual membership and delinquency process.

This batch is a multi-threaded batch. The multi-threading is based on membership ID and chunks for multi-threading are created based on numerical distribution of membership ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Outbound Message Type For Termination	Yes	<p>Used to indicate the outbound message type using which you want to create the termination outbound messages.</p> <div data-bbox="769 415 1409 537" style="border: 1px solid black; padding: 5px;"> <p>Note: You must specify an outbound message type where the outbound message business object is set to C1-MemberTermRequest.</p> </div>
Outbound Message Type For Cancellation	Yes	<p>Used to indicate the outbound message type using which you want to create the cancellation outbound messages.</p> <div data-bbox="769 705 1409 827" style="border: 1px solid black; padding: 5px;"> <p>Note: You must specify an outbound message type where the outbound message business object is set to C1-MemberCancRequest.</p> </div>
Source System	No	<p>Used to indicate the external source system to which you want to send the termination or cancellation outbound messages.</p> <div data-bbox="769 978 1409 1100" style="border: 1px solid black; padding: 5px;"> <p>Note: You must specify a source system which is already defined in the C1-SourceSystemLookup extendable lookup.</p> </div>
Membership Active Status	Yes	<p>Used to indicate the status code of the Active status of an individual membership.</p> <div data-bbox="769 1220 1409 1341" style="border: 1px solid black; padding: 5px;"> <p>Note: You must specify a status code which is already defined in the lifecycle of the C1-IndMembership business object.</p> </div>
Membership Awaiting Termination Status Reason	Yes	<p>Used to indicate the status reason which is specified in the Membership Termination Reason for Delinquency attribute of the delinquency process preference.</p> <div data-bbox="769 1493 1409 1614" style="border: 1px solid black; padding: 5px;"> <p>Note: You must specify a status reason which is already defined for the Active status of the C1-IndMembership business object.</p> </div>

Parameter Name	Mandatory (Yes or No)	Description
Membership Awaiting Cancellation Status Reason	Yes	Used to indicate the status reason which is specified in the Awaiting Membership Cancellation Reason attribute of the delinquency process preference. Note: You must specify a status reason which is already defined for the Active status of the C1-IndMembership business object.
Thread pool name	No	Used to specify the thread pool on which you want to execute the batch.
Chunk Size	Yes	Used to specify the number of outbound messages you want to create in each work unit.

Note: The system considers the delinquency process preference which is specified in the **Delinquency Process Field Mapping** option type of the **DELINPROC** feature configuration. For more information about the feature configuration, refer to the [DELINPROC Feature Configuration](#) section.

Delinquency Grace Period Business Rule

Oracle Revenue Management and Billing enables you to define business rules for determining the grace period during the delinquency management. The insurance company may offer different grace period to different individuals based on certain parameters, such as division, customer class, region, and jurisdiction in which the individual customer has availed for the health plan coverage. The system will then accordingly use the grace period to calculate the following:

- Trigger dates for the delinquency events
- Termination request date when you want to send the outbound message to the external system requesting to terminate the individual membership

The system considers the delinquency grace period business rules to determine the grace period only when the grace period source is set to **Algorithm** in the respective delinquency process type. If the grace period source is set to **Algorithm**, the system executes the **C1-DEGRBR** algorithm attached to the **Grace Period Derivation** system event of the delinquency process type when you execute the **C1-DPEVL** batch. For more information on how the grace period is derived, refer to the respective algorithm description in the [Delinquency Process Type Algorithms](#) section.

While defining the delinquency grace period business rule criteria, you can use the parameters which are defined on the following entities:

- Account
- Customer Class
- Delinquency Event Type
- Division
- Health Plan

- Health Product
- Membership
- Membership Person
- Policy
- Policy Plan

You can define, edit, and delete a delinquency grace period business rule through the **Business Rule** screen. While defining a delinquency grace period business rule, you need to set its category to **Delinquency Grace Period Business Rule** and specify the following details:

- **Criteria Description** – Used to indicate the business criteria that an individual membership must satisfy in order to derive the grace period from the business rule.
- **Effective Date** – Used to specify the date from when the business rule is effective.
- **Status** – Used to indicate the status of the business rule. The valid values are **Active** and **Inactive**.
- **Grace Period (Days)** – Used to specify the grace period in days.

You can view the delinquency grace period business rules through the **Business Rule** screen.

Delinquency Event Attribute Business Rule

Oracle Revenue Management and Billing enables you to define business rules for determining the waiting time (in days or months) before triggering a delinquency event in the delinquency process. The insurance company may offer different waiting time for different delinquency event types (such as sending letters, sending To Do notification, or initiating the termination request) based on certain parameters, such as division, customer class, region, and jurisdiction in which the individual customer has availed for the health plan coverage. The system will then accordingly use the wait days or the wait months and the day of month (depending on the wait period that is configured) to calculate the trigger dates for the delinquency events.

The system considers the delinquency event attributes business rules to determine the wait period only when the wait period source is set to **Algorithm** in the respective delinquency process type. If the wait period source is set to **Algorithm**, the system executes the **C1-DERWDBR** algorithm attached to the **Wait Days Derivation** system event of the delinquency process type when you execute the **C1-DPMON** batch. For more information on how the wait period is derived, refer to the respective algorithm description in the [Delinquency Process Type Algorithms](#) section.

While defining the delinquency event attributes business rule criteria, you can use the parameters which are defined on the following entities:

- Account
- Customer Class
- Delinquency Event Type
- Division
- Health Plan
- Health Product

- Membership
- Membership Person
- Policy
- Policy Plan

You can define, edit, and delete a delinquency event attributes business rule through the **Business Rule** screen. While defining a delinquency event attributes business rule, you need to set its category to **Delinquency Event Attributes Business Rule** and specify the following details:

- **Criteria Description** – Used to indicate the business criteria that an individual membership must satisfy in order to derive the wait period from the business rule.
- **Effective Date** – Used to specify the date from when the business rule is effective.
- **Status** – Used to indicate the status of the business rule. The valid values are Active and Inactive.
- **Wait Days** – Used to indicate the number of days for which you want to delay a delinquency event in a delinquency process.
- **Wait Month** – Used to indicate the number of months for which you want to delay a delinquency event in a delinquency process.
- **Wait Day of Month** – Used to indicate the day of the month when you want to trigger a delinquency event of a delinquency process.
- **Send Membership Level Notification** – This parameter is not applicable for the fully insured individual business.

You can view the delinquency event attributes business rules through the **Business Rule** screen.

Delinquency Termination Date Rule Business Rule

Oracle Revenue Management and Billing enables you to define business rules for determining the termination date rule during the delinquency management. The termination date rule is used to derive the termination date for a policy or individual membership. The insurance company may derive different termination date rule for different policies or individual memberships based on certain parameters, such as division, customer class, region, and jurisdiction, in which the group or individual customer has availed for the policy plan or health plan coverage. The system will then accordingly use the termination date rule to calculate the termination date for the delinquency event type that requests to terminate the following:

- Policies billed to the delinquent person or account
- Individual memberships billed to the delinquent account

The system always derives the termination date rule through a delinquency termination date rule business rule. The system executes the **C1-DERTBR** algorithm attached to the **Termination Date Rule Derivation** system event of the respective delinquency process type to derive the termination date rule for the delinquency event. For more information on how the termination date rule is derived, refer to the **C1-DERTBR** algorithm description in the [Delinquency Process Type Algorithms](#) section.

While defining the delinquency termination date rule business rule criteria, you can use the parameters which are defined on the following entities:

Line of Business	Source Entity	Supported Source Type
Fully Insured Group Health Insurance	Account	Field and Characteristic
	Customer Class	Field
	Division	Field
	Membership	Field and Characteristic
	Membership Person	Field and Characteristic
	Policy	Field and Characteristic
	Policy Plan	Field and Characteristic
Fully Insured Individual Health Insurance	Account	Field and Characteristic
	Customer Class	Field
	Division	Field
	Health Plan	Field and Characteristic
	Health Product	Field and Characteristic
	Membership	Field and Characteristic
	Membership Person	Field and Characteristic

Points to Note:

If the delinquency process is created at the person level and if an attribute of a division or customer class is used in the business rule criteria, the system will check whether the division or customer class is present on the person (i.e. on the parent customer or bill group, respectively) while evaluating the business rule criteria. If so, the system will use the details of the person to evaluate the business rule criteria. However, if the division or customer class is not present on the person, the system will use the details of any account where the person is added as the main customer to evaluate the business rule criteria.

If the delinquency process is created at the account level and if an attribute of a division or customer class is used in the business rule criteria, the system will use the details of the respective account to evaluate the business rule criteria.

You can define, edit, and delete a delinquency termination date rule business rule through the **Business Rule** screen. While defining a delinquency termination date rule business rule, you need to set its category to **Delinquency Termination Date Rule Business Rule** and specify the following details:

- **Criteria Description** – Used to indicate the business criteria that a policy or individual membership must satisfy in order to derive the termination date rule from the business rule.
- **Effective Date** – Used to specify the date from when the business rule is effective.
- **Status** – Used to indicate the status of the business rule. The valid values are **Active** and **Inactive**.
- **Termination Date Rule** – Used to indicate the termination date rule using which you want to calculate the termination date. The valid values are:

- Latest Billed Coverage End Date
- Latest Due Coverage End Date
- X days After Paid Through Date
- X Months After Paid Through Date
- Month End of Grace Period Start
- Month End of Termination Letter Creation
- Month End of Termination Request Event
- X days After Grace End Date
- X days After Termination Request Event

Once the business rule is created, the business rule ID is generated. While creating a business rule, the status is, by default, non-editable and set to **Active**. While editing a business rule, you can change its status from **Active** to **Inactive** and vice-versa. Note that you cannot edit or delete a business rule if it is referred in the system. You can view the delinquency termination date rule business rules through the **Business Rule** screen.

For more information on how the termination date rule is used to calculate the termination date, refer to the **C1-DETERMDT** algorithm description in the [Delinquency Event Activation](#) section.

Delinquency Miscellaneous Options Business Rule

Oracle Revenue Management and Billing enables you to define business rules for determining the following during the delinquency management:

- Whether the trigger dates of delinquency events should be recalculated when the hold on the delinquency process is released
- Whether the trigger dates of delinquency events should be recalculated when the delinquency process is resumed after cancellation
- Whether you want to send an outbound message to the enrollment system for terminating the individual memberships billed to the delinquent account immediately, at the end of the month when the termination event is triggered, or after X number of days (i.e. termination wait days) from the termination event trigger date
- The number of days that you want to add to the base date while calculating the termination date

The insurance company may offer different delinquency arrangements to different individuals based on certain parameters, such as division, customer class, region, and jurisdiction in which the individual customer has availed for the health plan coverage. The system will then accordingly use the business rule to calculate the following:

- Trigger dates for the delinquency events
- Termination request date when you want to send the outbound message to the external system requesting to terminate the individual membership
- Termination date for the delinquency event type that requests to terminate the individual memberships billed to the delinquent account

The system considers the delinquency miscellaneous options business rules to determine the following:

- The value for the **Trigger Date Recalculation On Hold Release** flag when the trigger date recalculation on hold release source is set to **Algorithm** in the respective delinquency process type. If the trigger date recalculation on hold release source is set to **Algorithm**, the system executes the **C1-DERRTDBR** algorithm attached to the **Trigger Date Recalculation Option Derivation** system event of the delinquency process type when you execute the **C1-DPMON** batch. For more information on how the value is derived for the **Trigger Date Recalculation On Hold Release** flag, refer to the respective algorithm description in the [Delinquency Process Type Algorithms](#) section.
- The value for the **Trigger Date Recalculation On Resume** flag when the trigger date recalculation on resume source is set to **Algorithm** in the respective delinquency process type. If the trigger date recalculation on resume source is set to **Algorithm**, the system executes the **C1-DERRTDBR** algorithm attached to the **Trigger Date Recalculation Option Derivation** system event of the delinquency process type when you execute the **C1-DPMON** batch. For more information on how the value is derived for the **Trigger Date Recalculation On Resume** flag, refer to the respective algorithm description in the **Delinquency Process Type Algorithms** section.

- The value for the **Trigger Termination Process** flag and the termination wait days when the termination reinstatement configuration source is set to Algorithm in the respective delinquency process type. If the termination reinstatement configuration source is set to **Algorithm**, the system executes the **C1-DERTROBR** algorithm attached to the **Termination and Reinstatement Configuration Derivation** system event of the delinquency process type when you execute the **C1-DPMON** batch. For more information on how the value is derived for the **Trigger Termination Process** flag, refer to the respective algorithm description in the [Delinquency Process Type Algorithms](#) section.
- The number of days that you want to add to the base date while deriving the termination date. The system executes the **C1-DETERMDT** algorithm attached to the **Event Activation** system event of the respective delinquency event type whenever the delinquency event is triggered manually or automatically on the trigger date. For more information on how the following parameters of the business rule are used to derive the termination date, refer to the respective algorithm description in the [Delinquency Event Activation](#) section:
 - Add Days to Coverage End Date
 - Add Days to Paid Through Date
 - Add Months to Paid Through Date
 - Add Days to Grace End Date
 - Add Days to Date of Processing of Termination Request

While defining the delinquency miscellaneous options business rule criteria, you can use the parameters which are defined on the following entities:

- Account
- Customer Class
- Delinquency Event Type
- Division
- Health Plan
- Health Product
- Membership
- Membership Person
- Policy
- Policy Plan

You can define, edit, and delete a delinquency miscellaneous options business rule through the **Business Rule** screen. While defining a delinquency miscellaneous options business rule, you need to set its category to **Delinquency Miscellaneous Options Business Rule** and specify the following details:

- **Criteria Description** – Used to indicate the business criteria that an individual membership must satisfy in order to derive the required parameters from the business rule.
- **Effective Date** – Used to specify the date from when the business rule is effective.
- **Status** – Used to indicate the status of the business rule. The valid values are **Active** and **Inactive**.

- **Trigger Date Recalculation On Hold Release** – Used to indicate whether the trigger dates of delinquency events should be recalculated when the hold on the delinquency process is released. The valid values are – **Yes** and **No**.
- **Trigger Date Recalculation On Resume** – Used to indicate whether the trigger dates of delinquency events should be recalculated when the delinquency process is resumed after cancellation. The valid values are – **Yes** and **No**.
- **Trigger Termination Process** – Used to indicate when you want the system to send an outbound message to the enrollment system for terminating the individual memberships billed to the delinquent account. The valid values are:
 - **Immediate** – Used when you want the system to send an outbound message for termination on the same day when the termination event is triggered.
 - **Month End** - Used when you want the system to send an outbound message for termination at the end of the month when the termination event is triggered.
 - **After X Days** - Used when you want the system to send an outbound message for termination after X number of days from the date when the termination event is triggered.
- **Termination Wait Days** – Used to specify the number of days after which an outbound message for termination should be sent to the enrollment system. This parameter is required only when the **Trigger Termination Process** parameter in the business rule is set to **After X Days**.
- **Add Days to Coverage End Date** – Used to specify the number of days that you want to add to coverage end date in order to derive termination date.
- **Add Days to Paid Through Date** - Used to specify the number of days that you want to add to the paid through date (of the account) in order to derive termination date.
- **Add Months to Paid Through Date** - Used to specify the number of months that you want to add to the paid through month (derived from the paid through date of the account) in order to derive termination date.
- **Add Days to Grace End Date** - Used to specify the number of days that you want to add to grace end date in order to derive termination date.
- **Add Days to Date of Processing of Termination Request** - Used to specify the number of days that you want to add to termination request date in order to derive termination date.

Note: The **Allow Automatic Reinstatement**, **Maximum Automatic Reinstatement Count**, **Trigger Reinstatement Process**, **Reinstatement Wait Days**, **Reinstatement Threshold (%)**, **Days Since Termination**, and **Member Notification Days** parameters in the delinquency miscellaneous options business rule are not applicable for the fully insured individual business.

You can view the delinquency miscellaneous options business rules through the **Business Rule** screen.

Hold Repricing

Until now, the system calculated the premium and benefit (if applicable) charges immediately through the **Repricing** feature whenever an individual membership is added, updated, canceled, terminated, or reinstated in ORMB. In the individual health insurance business, there might be situations wherein approved premium rates are not available for the future enrollment period, frequent updates are received for the memberships, or the data flow is delayed due to some maintenance activity in the source system. In all such situations, you may want to hold the repricing for the membership for a particular period.

Oracle Revenue Management and Billing enables you to capture the repricing after date for a membership. This helps you to hold the repricing for the membership until the specified date. Note that the hold repricing feature is offered and certified only for the fully insured individual business and not for the fully insured group business. To implement the hold repricing feature, the following changes are made in the system:

- A new field named **REPRC_AFTER_DT** is added in the **CI_MEMBERSHIP** table.
- A new tag named **repriceAfterDate** is introduced in the **C1-IndMembership** and **C1-HCInboundMessage** business object schema.
- A new tag named **repriceAfterDate** is introduced in the **C1-MembershipRepricing** business service schema.
- The **C1-HCINPROC** and **C1-INVRDTCIM** algorithm types are enhanced to support the hold repricing feature.

You can specify the repricing after date for an individual membership 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 **Membership** screen). The **repriceAfterDate** tag is not mandatory. If the value for the **repriceAfterDate** tag is received while creating an individual membership, the system checks whether the repricing after date of the individual membership is earlier than the inbound message processing date. If so, the system throws an appropriate message indicating that the repricing after date of the individual membership cannot be earlier than the inbound message processing date. However, if the repricing after date of an individual membership is equal to or later than the inbound message processing date, the system checks whether the repricing after date of an individual membership is later than the membership end date. If so, the system throws an appropriate message indicating that the repricing after date of the individual membership cannot be later than the membership end date. However, if the repricing after date of an individual membership is earlier than the membership end date, the system adds the individual membership to the health plan. The repricing after date (if any) is then shown in the **Main** section of the **Membership** screen. In addition, a log entry is created for the individual membership indicating that the repricing for the membership is on hold till a particular date. You can view the log entry in the **Log** tab of the **Membership** screen.

If the value for the **repriceAfterDate** tag is received while updating a membership, the system checks whether the repricing after date of the individual membership is earlier than the inbound message processing date. If so, the system throws an appropriate message indicating that the repricing after date of the individual membership cannot be earlier than the inbound message processing date. However, if the repricing after date of an individual membership is equal to or later than the inbound message processing date, the system checks the status of the individual membership. If the status of the individual membership is not set to **Canceled**, the system checks whether there is any invoice request created for the respective account in the non-final status (i.e. in a status other than **Processed** or **Canceled**). If so, the system checks whether the repricing after date is later than the invoice request processing date. If so, the system sets the invoice request processing date to repricing after date + 1. In addition, a log entry is created for the invoice request indicating that the invoice request is on hold till a particular date. You can view the log entry in the **Log** tab of the **Invoice Request** screen.

However, if the repricing after date is earlier than or equal to the invoice request processing date, the system does not override the invoice request processing date.

On executing the **C1-REPC1** batch, the system considers the repricing after date specified for the individual membership while processing the respective repricing entity detail records. If the repricing after date is equal to or later than the batch business date, the system considers only those repricing entity detail records of the individual membership where the category of the pricing rule types is set to **Benefit** and accordingly creates the repricing requests for the benefit specific price items. Only when the repricing after date is earlier than the batch business date, the system considers all the repricing entity detail records of the respective individual membership which are in the **Pending** status and accordingly creates a repricing request for each individual membership, pricing rule, and effective date combination.

Maintaining Address Status

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.

Points to 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**.

Prorate Membership Benefit and Corresponding Sponsor Charges

Until now, the system prorated the group and individual membership premium charges when the membership was enrolled or terminated in middle of the billing cycle. For the fully insured group business, the system prorated the membership premium using the proration rules defined on the fully insured policy plan. However, for the fully insured individual business, the system prorated the membership premium using the daily proration mechanism.

Now, Oracle Revenue Management and Billing also enables you to prorate the membership benefit charges on the eligible account and their corresponding benefit sponsor charges on the sponsor account. For the fully insured group business, the system allows you to prorate the membership premium benefit charges using the proration rules defined on the fully insured policy plan and prorate the membership non-premium benefit charges using the daily proration mechanism. A new option named **Premium Benefit** is introduced while defining and editing a benefit sub type. If the **Premium Benefit** option is selected in a benefit sub type, the system will consider the benefits created using the benefit sub type as the membership premium benefits and will prorate the membership premium benefits (if required) using the proration rules defined on the fully insured policy plan. Note that if the proration rules are not defined on the fully insured policy plan, the system will prorate the membership premium benefits using the daily proration mechanism.

However, if the **Premium Benefit** option is not selected in a benefit sub type, the system will consider the benefits created using the benefit sub type as the membership non-premium benefits and will prorate the membership non-premium benefits (if required) using the daily proration mechanism. While defining a benefit sub type, by default, the **Premium Benefit** option is not selected. If you select the **Premium Benefit** option, then the **Charge Benefit Sponsor** option is removed from the user interface. This is because the group membership premium benefits are not sponsored by a health insurance exchange. If the **Premium Benefit** option is not selected and the **Charge Benefit Sponsor** option is selected, the system will prorate the benefit sponsor charges using the daily proration mechanism.

For the fully insured individual business, the system will prorate the membership benefit charges and their corresponding benefit sponsor changers using the daily proration mechanism.

Points to Note:

The **Premium Benefit** option in the **Benefit Sub Type** screen is applicable only for the fully insured group business.

The end date of benefit should be sent through a health care inbound message or through a membership repricing request whenever the group or individual membership is terminated.

The system decides whether to prorate the membership premium or non-premium benefit charges of a price item using the **Proration Required** option defined corresponding to the price item in the respective benefit pricing rule type. If the **Proration Required** option is selected corresponding to a price item, the system will prorate the charges of the respective price item whenever required during membership enrolment or termination. However, if the **Proration Required** option is not selected corresponding to a price item, the system will not prorate the charges of the respective price item during membership enrolment or termination.

Note that if you select the **Proration Required** option corresponding to any price item in a benefit pricing rule type, you need to attach an algorithm created using the **C1-BNFTPRCF** algorithm type to the **Pricing Rule Proration Amount Calculation** system event of the benefit pricing rule type.

Zero-Amount Membership Benefit Charges

Oracle Revenue Management and Billing enables you to receive the full or partial snapshot of benefit charges for a group or individual membership through a health care inbound message or through a membership repricing request. A new option type named **Benefit Full Snapshot Upload** is introduced in the **C1-ASOBLNG** feature configuration. It enables the system to determine whether the full or partial snapshot of benefit charges is received for a group or individual membership through a health care inbound message or through a membership repricing request. You can set the **Benefit Full Snapshot Upload** option type to **Y** or **N**. If you set the **Benefit Full Snapshot Upload** option type to **Y**, the system behaves in the following manner in the respective scenarios:

Scenario	Existing Benefit Data			New Snapshot (Received through an Inbound Message)			Comments
	Benefit Record	Benefit Coverage	Benefit Amount	Benefit Record	Benefit Coverage	Benefit Amount	
S1	B1	Jan – Mar	100	B1	Jan – Mar	90	The system will update the B1 (Jan-Mar), B2 (Apr-Jun), and B3 (Jul-Dec) benefit records of the membership.
	B2	Apr – Jun	100	B2	Apr – Jun	110	
	B3	Jul – Dec	200	B3	Jul – Dec	120	
S2	B1	Jan – Mar	50	B3	Jan – Mar	50	The system will update the B1 (Jan-Mar) and B3 (Jul-Dec) benefit records of the membership. In
	B2	Apr – Jun	90		Jul – Dec	120	
	B3	Jul – Dec	150				

Scenario	Existing Benefit Data			New Snapshot (Received through an Inbound Message)			Comments
	Benefit Record	Benefit Coverage	Benefit Amount	Benefit Record	Benefit Coverage	Benefit Amount	
							addition, it will change the status of the B2 (Apr-Jun) benefit record to Inactive .
S3	B1	Jan – Mar	100	B1	Jan – Mar	100	The system will update the B1 (Jan-Mar), B2 (Apr-Jun), and B3 (Jul-Dec) benefit records of the membership.
	B2	Apr – Jun	100	B2	Apr – Jun	0	
	B3	Jul – Dec	200	B3	Jul – Dec	200	
S4	B1	Jan – Mar	100	B11	Jan – Feb	90	The system will update the B3 (Jul-Dec) benefit record of the membership. However, the system will change the status of B1 (Jan-Mar) and B2 (Apr-Jun) benefit records to Inactive and will create two new benefit records – B11 (Jan-Feb) and B12 (Mar-Jun) with the Active status.
	B2	Apr – Jun	100	B12	Mar – Jun	110	
	B3	Jul – Dec	200	B3	Jul – Dec	120	

Note: If you set the **Benefit Full Snapshot Upload** option type to **Y**, the system will update the existing benefit records when the benefit coverage period matches.

However, if you set the **Benefit Full Snapshot Upload** option type to **N**, the system behaves in the following manner in the respective scenarios:

Scenario	Existing Benefit Data			New Snapshot (Received through an Inbound Message)			Comments
	Benefit Record	Benefit Coverage	Benefit Amount	Benefit Record	Benefit Coverage	Benefit Amount	
S1	B1	Jan – Mar	100	B1	Jan – Mar	90	The system will change the status of the B1 (Jan-Mar), B2 (Apr-Jun), and B3
	B2	Apr – Jun	100	B2	Apr – Jun	110	
	B3	Jul – Dec	200	B3	Jul – Dec	120	

Scenario	Existing Benefit Data			New Snapshot (Received through an Inbound Message)			Comments
	Benefit Record	Benefit Coverage	Benefit Amount	Benefit Record	Benefit Coverage	Benefit Amount	
							(Jul-Dec) benefit records to Inactive . The system will then create three new benefit records - B1 (Jan-Mar), B2 (Apr-Jun), and B3 (Jul-Dec) with the Active status.
S2	B1	Jan – Mar	100	B1	Jan - Dec	300	The system will change the status of the B1 (Jan-Mar), B2 (Apr-Jun), and B3 (Jul-Dec) benefit records to Inactive . The system will then create one new benefit record – B1 (Jan-Dec) with the Active status.
	B2	Apr – Jun	100				
	B3	Jul – Dec	200				
S2	B1	Jan – Mar	100	B1	Jan - Jun	300	The system will change the status of the B1 (Jan-Mar), B2 (Apr-Jun), and B3 (Jul-Dec) benefit records to Inactive . The system will then create two new benefit records – B1 (Jan-Jun) and B2 (Jul-Dec) with the Active status.
	B2	Apr – Jun	100	B2	Jul-Dec	0	
	B3	Jul – Dec	200				

Note: If you set the **Benefit Full Snapshot Upload** option type to **N**, the system will always inactivate all the existing benefit records of the membership and will create the new benefit records for the membership in the **Active** status.

A new option type named **Zero Dollar Benefit Charge** is introduced in the **C1-ASOBLNG** feature configuration. It enables the system to determine whether the billable charge for zero benefit amount should be created or not. You can set the **Zero Dollar Benefit Charge** option type to **Y** or **N**. If you set the **Zero Dollar Benefit Charge** option type to **Y**, the system will create a zero-amount benefit charge for the membership. However, if you set the **Zero Dollar Benefit Charge** option type to **N**, the system will not create a zero-amount benefit charge for the membership.

Additional Repricing Entity Detail Records

Until now, while creating repricing entity detail records for an audit event, the system used to create two repricing entity detail records for each membership and pricing rule type combination – one with the effective date as the membership start date and another with the effective date as the auditable element effective date. In addition, the system used to set the **Message_PARAM4** column to **Y** corresponding to the latter repricing entity detail record (with the effective date as the auditable element effective date) so that the system creates a repricing request only for the latter repricing entity detail record and not for the former repricing entity detail record.

A new option type named **Additional Repricing** is added in the **C1-ASOBLNG** feature configuration. It enables the system to determine whether you want to create repricing entity detail records for all auditable elements (such as fields, business statuses, and characteristics) of the membership, member person, or person that are updated within the membership period. You can set the **Additional Repricing** option type to either of the following:

- **Y** – Used when you want to create repricing entity detail records for all auditable elements (such as fields, business statuses, and characteristics) of the membership, member person, or person that are updated within the membership period.
- **N** – Used when you want to create repricing entity detail records in the traditional manner (i.e. one with the effective date as the membership start date and another with the effective date as the auditable element effective date).

If you do not specify the value for this option type, the system, by default, sets it to **N**. Note that this option type is only applicable for the fully insured individual business.

Let us assume the following:

- M1 membership is effective from 01-Jan-2023 to 31-Dec-2023 on the HP1 health plan
- AGE_BASED1 and BENEFIT_1 pricing rule types are associated with the HP1 health plan
- The following auditable elements at the membership or member person level are already updated till 01-08-2023:

Element Name	Element Value	Modified On
benefitAmount	Field	10-Jan-2023
billLevel1	Field	20-Feb-2023
emailAddress	Field	22-Mar-2023
C1SMOKER	Characteristic	04-Apr-2023

Element Name	Element Value	Modified On
C1TOBCC	Characteristic	05-May-2023
INACTVAL	Characteristic	01-Aug-2023

Now, if the **Additional Repricing** option type is set to **Y** and you update the **INACTVAL** characteristic of the M1 membership on 01-Aug-2023, the system will create the following repricing entity detail records for the audit event:

- M1, AGE_BASED1, 01-Jan-2023 (Message_PARAM4 = N)
- M1, BENEFIT_1, 01-Jan-2023 (Message_PARAM4 = N)
- M1, AGE_BASED1, 10-Jan-2023 (Message_PARAM4 = N)
- M1, BENEFIT_1, 10-Jan-2023 (Message_PARAM4 = N)
- M1, AGE_BASED1, 20-Feb-2023 (Message_PARAM4 = N)
- M1, BENEFIT_1, 20-Feb-2023 (Message_PARAM4 = N)
- M1, AGE_BASED1, 22-Mar-2023 (Message_PARAM4 = N)
- M1, BENEFIT_1, 22-Mar-2023 (Message_PARAM4 = N)
- M1, AGE_BASED1, 04-Apr-2023 (Message_PARAM4 = N)
- M1, BENEFIT_1, 04-Apr-2023 (Message_PARAM4 = N)
- M1, AGE_BASED1, 05-May-2023 (Message_PARAM4 = N)
- M1, BENEFIT_1, 05-May-2023 (Message_PARAM4 = N)
- M1, AGE_BASED1, 01-Aug-2023 (Message_PARAM4 = Y)
- M1, BENEFIT_1, 01-Aug-2023 (Message_PARAM4 = Y)

The system will create repricing requests only for the repricing entity detail records where the **Message_PARAM4** column is set to **Y**.

Allocate Advance Deposit for Delinquent Customers

In the fully insured group business, the group customers may opt to pay the advance deposit upfront so that it can be used in case of delinquency to settle unpaid amount. The group customer may pay the advance deposit at the parent customer or policy level. Oracle Revenue Management and Billing enables you to park the advance deposit received from the group customers at the parent customer or policy level.

If you create a bill with charges from multiple policies, then the system should receive advance deposit at the parent customer level. The system will then use the advance deposit to pay unpaid bills of all the policies of the parent customer where he is the policy holder. The system enables you to create an account at the parent customer level to park the advance deposit. However, if you create a bill with charges from a single policy, then the system should receive advance deposit at the policy level. The system will then use the advance deposit to pay unpaid bills of the respective policy. The system enables you to create a policy-specific account at the parent customer level to park the advance deposit.

Maintaining Advance Deposit Details

The system enables you to provide the advance deposit details for a parent customer through a health care inbound message or through a membership repricing request. Similarly, the system enables you to provide the advance deposit details for a policy from the **Policy** screen or through a health care inbound message. You need to provide the following advance deposit details at the parent customer level:

- **Advance Deposit Level** – Used to indicate whether you want to park the advance deposit received from the group customer at the parent customer or policy level. The valid values are:
 - **PG** – Used when you want to park the advance deposit at the parent customer level.
 - **POLI** - Used when you want to park the advance deposit at the policy level.

The advance deposit level is only applicable for the person whose person type is set to **Parent Customer**. If you do not specify the advance deposit level, then the parent customer is not eligible for the advance deposit either at the parent customer or policy level.

- **Advance Deposit Amount** – Used to specify the advance deposit amount. It is required when the advance deposit level is set to **PG**.
- **Advance Deposit Grace Days** – Used to specify the grace days. The system then uses the grace days to derive the grace period within which the advance deposit should be received from the group customer. While calculating the grace period, the system derives the earliest start date among all the policies where the parent customer is the policy holder and considers it as the grace start date. Once the grace start date is derived, the system calculates the grace end date (i.e. grace start date + grace days). For example, if a parent customer is the policy holder of two policies – P1 (01-Jan-2022 to 31-Dec-2022) and P2 (01-Jun-2022 to 31-Dec-2022) and advance deposit grace days is set to 6 days, then the system considers 01-Jan-2022 as the grace start date (as it is the earliest start date) and sets the grace end date to 07-Jan-2022. The advance deposit grace days is required when the advance deposit level is set to **PG**.
- **Advance Deposit Account Details** – Used to specify the account details for the parent customer when the advance deposit is parked at the parent customer level.
 - **Account Identifier Type** – Used to specify the account identifier type.
 - **Account Identifier** – Used to specify the account identifier.
 - **Account Relationship Type** – Used to specify the account relationship type using which you want to associate the parent customer with the account. Here, you must specify an account relationship type which is specified in the **Advance Deposit Account Relationship Type** attribute of the delinquency process preference.

Note: The system considers the delinquency process preference which is specified in the **Delinquency Process Field Mapping** option type of the **DELINPROC** feature configuration. For more information about the feature configuration, refer to the [DELINPROC Feature Configuration](#) section.

The respective tags are added to provide the above advance deposit details at the parent customer level in the **C1-HCInboundMessage** business object schema and in the **C1-MembershipRepricing** business service schema. For more information, refer to the [Inbound Message](#) and [C1-MembershipRepricing Business Service](#) section, respectively.

If the advance deposit level of a policy holder (i.e. parent customer) is set to **POLI**, you need to specify the following advance deposit details at the policy level:

- **Advance Deposit Applicability** – Used to indicate whether the advance deposit is applicable for the policy. The valid values are **Y** and **N**. If you do not specify the value, the system, by default, sets it to **N**.
- **Advance Deposit Amount** - Used to specify the advance deposit amount. It is required when the advance deposit applicability is set to **Y**.
- **Advance Deposit Grace Days** - Used to specify the grace days. The system then uses the grace days to derive the grace period within which the advance deposit should be received from the group customer for the policy. While calculating the grace period, the system considers the policy start date as the grace start date. Once the grace start date is derived, the system calculates the grace end date (i.e. grace start date + grace days). For example, if a parent customer is the policy holder of the P1 policy (01-Jan-2022 to 31-Dec-2022) and advance deposit grace days is set to 6 days, then the system considers 01-Jan-2022 as the grace start date and sets the grace end date to 07-Jan-2022. The advance deposit grace days is required when the advance deposit applicability is set to **Y**.
- **Advance Deposit Account Identifier Type** – Used to specify the account identifier type. It is required when the advance deposit applicability is set to **Y**.
- **Advance Deposit Account Identifier** - Used to specify the account identifier. Here, you must specify an account of the policy holder where the account relationship type is set to the value that is specified in the **Advance Deposit Account Relationship Type** attribute of the delinquency process preference. It is required when the advance deposit applicability is set to **Y**.

Note: The system considers the delinquency process preference which is specified in the **Delinquency Process Field Mapping** option type of the **DELINPROC** feature configuration. For more information about the feature configuration, refer to the [DELINPROC Feature Configuration](#) section.

The respective tags are added to provide the above advance deposit details at the policy level in the **C1-HCInboundMessage** business object schema. For more information, refer to the [Inbound Message](#) section.

Applying Advance Deposit Payments

You can apply the advance deposit payment on the advance deposit account of the parent customer from the user interface or through an inbound web service. Note that the system distinguishes an account as the advance deposit account when the parent customer is associated with the account using the account relationship type which is specified in the **Advance Deposit Account Relationship Type** attribute of the delinquency process preference. While applying the advance deposit payment against the advance deposit account, you must specify a match type where an algorithm created using the **C1-MATSATYP** algorithm type is attached to the **Payment Distribution Override** algorithm spot.

In the **C1-MATSATYP** algorithm, you must specify a contract type that is specified in the **Advance Deposit Contract Type** attribute of the delinquency process preference. The system then applies the advance deposit payment against an active advance deposit contract of the specified contract type on the advance deposit account. If an active contract of the specified contract type does not exist on the advance deposit account, the system creates an advance deposit contract using the specified contract type and then applies the advance deposit payment against the advance deposit contract.

Note: The system considers the delinquency process preference which is specified in the **Delinquency Process Field Mapping** option type of the **DELINPROC** feature configuration. For more information about the feature configuration, refer to the [DELINPROC Feature Configuration](#) section.

Monitoring Advance Deposit of a Parent Customer

A new batch control named **C1-ADMON** is introduced in this release. It considers all the parent customers where the advance deposit level is set to **PG** or **POLI**. For a parent customer where advance deposit level is set to **PG**, the batch checks whether any customer contact is created using the given customer contact type for the given customer contact class where the respective person ID is stored as a characteristic on the customer contact. The system considers the characteristic type which is specified in the **Person ID Characteristic for Inbound Log** option type of the **C1-ASOBLING** feature configuration. If a customer contact exists for the parent customer, the batch does not consider the parent customer for further processing.

However, if a customer contact does not exist for the parent customer, the batch derives the advance deposit amount and advance deposit grace days from the parent customer level. It then calculates the grace end date and checks whether the grace end date is earlier than or equal to batch business date. If so, the system derives the advance deposit account relationship type and advance deposit contract type from the delinquency process preference. The system then derives the parent customer's advance deposit account and advance deposit contract using the advance deposit account relationship type and advance deposit contract type, respectively. The batch then checks whether the payments received against the advance deposit contract exceeds the advance deposit amount. If the sum of payments does not exceed the advance deposit amount, the system creates a customer contact using the given customer contact type for the given customer contact class. In addition, the system considers the characteristic type that is specified in the **Person ID Characteristic for Inbound Log** option type of the **C1-ASOBLING** feature configuration and then stamps the person ID using the respective characteristic type on the customer contact.

However, if the grace end date is later than the batch business date or if the sum of payments exceeds the advance deposit amount, the system does not consider the parent customer for further processing.

Similarly, for a parent customer where advance deposit level is set to **POLI**, this batch fetches all the policies where the parent customer is the policy holder and where the advance deposit applicability is set to **Y**. The system considers those policies where the parent customer is associated using the policy person role that is specified in the **Parent Customer Policy Person Role** option type of the **C1-ASOBLING** feature configuration.

For each such policy, this batch checks whether any customer contact is created using the given customer contact type for the given customer contact class where the respective policy ID is stored as a characteristic on the customer contact. The system considers the characteristic type which is specified in the **Policy Characteristic Type for Customer Contact** attribute of the delinquency process preference. If a customer contact exists for the policy, the batch does not consider the policy for further processing.

However, if a customer contact does not exist for the policy, the batch derives the advance deposit amount and advance deposit grace days from the policy. It then calculates the grace end date and checks whether the grace end date is earlier than or equal to batch business date. If so, the system derives the advance deposit account linked to the policy and the advance deposit contract using the advance deposit contract type specified in the delinquency process preference. The batch then checks whether the payments received against the advance deposit contract exceeds the advance deposit amount. If the sum of payments does not exceed the advance deposit amount, the system creates a customer contact using the given customer contact type for the given customer contact class. In addition, the system considers the characteristic type that is specified in the **Policy Characteristic Type for Customer Contact** attribute of the delinquency process preference and then stamps the policy ID using the respective characteristic type on the customer contact.

However, if the grace end date is later than the batch business date or if the sum of payments exceeds the advance deposit amount, the system does not consider the policy for further processing.

Note: The system considers the delinquency process preference which is specified in the **Delinquency Process Field Mapping** option type of the **DELINPROC** feature configuration. For more information about the feature configuration, refer to the [DELINPROC Feature Configuration](#) section.

This batch also derives the preferred contact method for the customer contact. While deriving the preferred contact method, the batch first fetches the main customer's bill route type on the advance deposit account and then fetches the bill routing method of the bill route type. Once the bill routing method is derived, this batch considers the preference which is specified in the **Routing Method – Contact Method Mapping** option type of the **DELINPROC** feature configuration. For more information about the feature configuration, refer to the [DELINPROC Feature Configuration](#) section.

It then checks whether any preferred contact method is mapped to the bill routing method in the preference. If the preferred contact method is mapped to the bill routing method in the preference, the system stamps it on the customer contact. However, if the preferred contact method is not mapped to the bill routing method in the preference, the system considers the default contact method as the preferred contact method and stamps it on the customer contact.

This batch is a multi-threaded batch. The multi-threading is based on person ID and chunks for multi-threading are created based on numerical distribution of person ID. You can specify the following parameters while executing this batch:

Parameter Name	Mandatory (Yes or No)	Description
Customer Contact Type	Yes	Used to indicate the customer contact type using which you want to create the customer contact. Note: You must specify a customer contact type with the appropriate letter template depending on the type of letter you want to send to the customer.
Customer Contact Class	Yes	Used to indicate the customer contact class to which the customer contact type belongs.
Default Contact Method	Yes	Used to indicate the contact method that you want to use when the preferred contact method cannot be derived for the routing method from the preference.
Chunk Size	Yes	Used to specify the number of persons you want to process in each work unit.
Thread Pool Name	No	Used to specify the thread pool on which you want to execute the batch.

Settling Unpaid Dues Against Advance Deposit through a Delinquency Process

Oracle Revenue Management and Billing enables you to settle the unpaid bill amount of a parent customer using the advance deposit available at the parent customer or policy level through a delinquency process. For example, an advance deposit of \$1000 is received from the parent customer through multiple payments as shown below:

Advance Deposit Payments						
Payment Event ID	Payment ID	Payment Date	Payment Amount	Payment Segment (SA1)	Match Event	Match Event Status
PE1	P1	01-Jan-2020	200	-200	ME1	Open
PE2	P2	05-Jan-2020	200	-200	ME2	Open
PE3	P3	10-Jan-2020	200	-200	ME3	Open
PE4	P4	11-Jan-2020	200	-200	ME4	Open
PE5	P5	15-Jan-2020	200	-200	ME5	Open
Actual Deposit				-1000		

Advance Deposit Payments						
Payment Event ID	Payment ID	Payment Date	Payment Amount	Payment Segment (SA1)	Match Event	Match Event Status
SA1 Contract Balance				-1000		

An advance deposit delinquency process is created for the parent customer which includes two bills – B1 and B2 with the bill segments as shown below:

Bill	Bill Segment	Amount
B1	-	300
	BSEG1 (SA2)	200
	BSEG2 (SA3)	100
B2	-	400
	BSEG1 (SA2)	250
	BSEG2 (SA3)	150

When the apply advance deposit delinquency event is triggered, the system will create four debit adjustments and one credit adjustment as shown in the below table. Note that the system will create the debit adjustments against the payment segments on the advance deposit contract using the debit adjustment type given in the **C1-APPADVDEP** algorithm. While creating the debit adjustments, the system will consider the payments in the ascending order of the payment date. In addition, the system will create one credit adjustment using the credit adjustment type given in the **C1-APPADVDEP** algorithm. This credit adjustment would be matched later in the delinquency process.

Advance Deposit Payments							Adjustment		
Payment Event ID	Payment ID	Payment Date	Payment Amount	Payment Segment (SA1)	Match Event	Match Event Status	Adjustment	Match Event	Match Event Status
PE1	P1	01-Jan-2020	200	-200	ME1	Open	200 (SA1)	ME1	Balanced
PE2	P2	05-Jan-2020	200	-200	ME2	Open	200 (SA1)	ME2	Balanced

Advance Deposit Payments							Adjustment		
Payment Event ID	Payment ID	Payment Date	Payment Amount	Payment Segment (SA1)	Match Event	Match Event Status	Adjustment	Match Event	Match Event Status
PE3	P3	10-Jan-2020	200	-200	ME3	Open	200 (SA1)	ME3	Balanced
PE4	P4	11-Jan-2020	200	-200	ME4	Open	100 (SA1)	ME4	Open
PE5	P5	15-Jan-2020	200	-200	ME5	Open			
Actual Deposit				-1000					
SA1 Contract Balance				-1000					
Credit Adjustment (SA1)							-700	-	-

The system will then create an offset request using the offset request type given in the **C1-APPADVDEP** algorithm. In the offset request, the system will include the credit adjustment (-\$700) and two bills – B1 and B2 for which the delinquency process is created. The system will then automatically process the offset request and create the offset request adjustments as shown in the below table. Note that the advance deposit is applied in the ascending order of the bill date. If two or more bills have the same bill date, the system applies the advance deposit in the ascending order of the unpaid amount (i.e. lowest unpaid amount is paid first).

Offset Request										
Credit Items					Debit Items					
Adjustment	Offset Request Adjustment	Offset Adjustment Amount	Match Event	Match Event Status	Bill	Bill Segment	Offset Request Adjustment	Offset Adjustment Amount	Match Event	Match Event Status
CA1 (SA1)	OA11	300	ME10	Balanced	B1	BSEG1 (SA2)	OA2 (SA2)	-200	ME6	Balanced

Offset Request										
Credit Items					Debit Items					
Adjustment	Offset Request Adjustment	Offset Adjustment Amount	Match Event	Match Event Status	Bill	Bill Segment	Offset Request Adjustment	Offset Adjustment Amount	Match Event	Match Event Status
						BSEG2 (SA3)	OA3 (SA3)	-100	ME7	Balanced
	OA12	400	ME10	Balanced	B2	BSEG1 (SA2)	OA4(SA2)	-250	ME8	Balanced
						BSEG2 (SA3)	OA5(SA3)	-150	ME9	Balanced

Once the offset request adjustments are created, the system creates a customer contact depending on whether the advance deposit is fully or partially utilized to pay the delinquent bills. The customer contact is created for the parent customer or for the main customer of the account depending on whether the delinquency process is created at the person or account level. In addition, the system stamps the delinquency process ID as a characteristic on the customer contact and on the advance deposit payments (used for offsetting; in this case P1, P2, P3, and P4). The system considers the characteristic type given in the **C1-APPADVDEP** algorithm. For more information about the algorithm, refer to [Delinquency Event Type](#) section.

The status of the delinquency process will remain as **Delinquency In Progress**. On executing the **C1-DPMON** batch, the system checks whether the delinquency process meets the cancel criteria. If so, the system will change the status of the delinquency process to **Canceled**.

For more information about the advance deposit delinquency process, refer to the [Advance Deposit Delinquency Process](#) section.

Notification on Receiving Advance Deposit

Oracle Revenue Management and Billing enables you to send notification via a customer contact when the advance deposit payment is received fully for a customer at the parent customer or policy level. To enable this feature, you need to attach an algorithm created using the **C1-NOTADREC** algorithm type to the **Payment Freeze** system event of the parent customer's customer class.

This algorithm checks whether the payment is made on a contract which is created using a contract type specified in the **Advance Deposit Contract Type** attribute of the delinquency process preference. If the payment is not made on a contract of the advance deposit contract type, no action takes place while freezing the payment. However, if the payment is made on a contract of the advance deposit contract type, this algorithm checks whether the advance deposit contract is linked to any policy. If so, the system derives the advance deposit amount specified on the policy and then checks whether the sum of payments received on the advance deposit contract is greater than or equal to the advance deposit amount specified on the policy. If so, the system creates a customer contact using the given customer contact type for the given customer contact class. In addition, the system stamps the policy ID as a characteristic on the customer contact using the characteristic type which is specified in the **Policy Characteristic Type for Customer Contact** option type of the **DELINPROC** feature configuration. However, if the sum of payments received on the advance deposit contract is less than the advance deposit amount, no action takes place while freezing the payment.

If the advance deposit contract is not linked to any policy, the system derives the account for which the advance deposit contract is created. The system then derives the person who is associated with the account using the relationship type which is specified in the **Advance Deposit Account Relationship Type** attribute of the delinquency process preference. Once the person is derived, the system derives the advance deposit amount specified at the parent customer level. The system then checks whether the sum of payments received on the advance deposit contract is greater than or equal to the advance deposit amount specified at the parent customer level. If so, the system creates a customer contact using the given customer contact type for the given customer contact class. However, if the sum of payments received on the advance deposit contract is less than the advance deposit amount, no action takes place while freezing the payment.

This algorithm also derives the preferred contact method for the customer contact. While deriving the preferred contact method, this algorithm first fetches the main customer's bill route type on the advance deposit account and then fetches the bill routing method of the bill route type. Once the bill routing method is derived, the system considers the preference which is specified in the **Routing Method – Contact Method Mapping** option type of the **DELINPROC** feature configuration. For more information about the feature configuration, refer to the [DELINPROC Feature Configuration](#) section.

It then checks whether any preferred contact method is mapped to the bill routing method in the preference. If the preferred contact method is mapped to the bill routing method in the preference, the system stamps it on the customer contact. However, if the preferred contact method is not mapped to the bill routing method in the preference, the system considers the default contact method as the preferred contact method and stamps it on the customer contact.

Note: The system considers the delinquency process preference which is specified in the **Delinquency Process Field Mapping** option type of the **DELINPROC** feature configuration. For more information about the feature configuration, refer to the [DELINPROC Feature Configuration](#) section.

This algorithm contains the following parameters:

- **Customer Contact Type** - Used to indicate the customer contact type using which you want to create the customer contact. Here, you must specify a customer contact type with the appropriate letter template depending on the type of letter you want to send to the customer.
- **Customer Contact Class** - Used to indicate the customer contact class to which the customer contact type belongs.
- **Default Contact Method** - Used to indicate the contact method that you want to use when the preferred contact method cannot be derived for the routing method from the preference.

All these parameters are mandatory.

Notification on Advance Deposit Cancellation

Oracle Revenue Management and Billing enables you to send notification via a customer contact when the payment on the advance deposit contract is canceled. To enable this feature, you need to attach an algorithm created using the **C1-NOTADCAN** algorithm type to the **Payment Cancellation** system event of the parent customer's customer class.

This algorithm checks whether the payment is made on a contract which is created using a contract type specified in the **Advance Deposit Contract Type** attribute of the delinquency process preference. The system considers the delinquency process preference which is specified in the **Delinquency Process Field Mapping** option type of the **DELINPROC** feature configuration. For more information about the feature configuration, refer to the [DELINPROC Feature Configuration](#) section.

If the payment is not made on a contract of the advance deposit contract type, no action takes place while canceling the payment. However, if the payment is made on a contract of the advance deposit contract type, this algorithm checks whether the cancel reason specified while canceling a payment is listed in the **Advanced Deposit Payment Cancel Reason List** parameter. If so, the system creates a customer contact using the given customer contact type for the given customer contact class. If the advance deposit contract is linked to a policy, the system stamps the policy ID as a characteristic on the customer contact using the characteristic type which is specified in the **Policy Characteristic Type for Customer Contact** option type of the **DELINPROC** feature configuration.

However, if the cancel reason specified while canceling a payment is not listed in the **Advanced Deposit Payment Cancel Reason List** parameter, no action takes place while canceling the payment. If the advance deposit payment is already used to settle an overdue bill through a delinquency process, the system does the following:

- Cancels the debit adjustments created against the advance deposit payment.
- Cancels the credit adjustments used to offset the overdue bills through a delinquency process.

- Unapplies or cancels the offset request. If the number of transfer or offset adjustments does not exceed the online record process limit (defined in the **C1-DFRUNAPLY** algorithm), the system cancels the frozen transfer or offset adjustments immediately and then changes the status of the offset request to **Unapplied Offset**. However, if the number of transfer or offset adjustments exceeds the online record process limit (defined in the **C1-DFRUNAPLY** algorithm), the system changes the status of the offset request to **Defer Unapplied**.

Points to Note:

On executing the **Offset Request Periodic Monitor (C1-OFSRQ)** batch, the system considers the offset requests which are in the **Defer Unapplied** status. The system then cancels the frozen transfer or offset adjustments of the offset request and changes the status of the offset request to **Unapplied Offset**.

The system then executes the **C1-CDPPAYCAN** algorithm attached to the **Payment Cancellation** system event of the parent customer's customer class. It changes the status of the delinquency process. For more information about the algorithm, refer to the [Resuming a Delinquency Process on Payment Cancellation](#) section.

Always ensure that you attach the following algorithms to the **Payment Cancellation** system event in the specified sequence:

1. C1-NOTADCAN
2. C1-CDPPAYCAN

This algorithm contains the following parameters:

- **Customer Contact Type** - Used to indicate the customer contact type using which you want to create the customer contact. Here, you must specify a customer contact type with the appropriate letter template depending on the type of letter you want to send to the customer.
- **Customer Contact Class** - Used to indicate the customer contact class to which the customer contact type belongs.
- **Default Contact Method** - Used to indicate the contact method that you want to use when the preferred contact method cannot be derived for the routing method from the preference.
- **Advanced Deposit Payment Cancel Reason List** – Used to specify a comma-separated list of payment cancel reasons that the system should consider while notifying customers on payment cancelation. The system uses this parameter to distinguish between advance deposit transfer and advance deposit cancellation. You can specify maximum five comma-separated values for this parameter. Here, you must specify a payment cancel reason which is already defined in the system.
- **Offset Request Defer Unapply Status Code** – Used to specify the status to which you want to transition the offset request when an advance deposit payment is canceled. You must specify a status code which is already defined in the lifecycle of the **C1-OffsetRequest** business object.
- **Offset Request Defer Unapply Status Reason** – Used to specify the reason why you want to unapply or cancel the offset request. You must specify a reason which is already defined for the **Defer Unapplied** status of the **C1-OffsetRequest** business object in the **Status Reason** screen.

All these parameters are mandatory.

Accounts Payable (Refund Adjustments) Extraction

Oracle Revenue Management and Billing enables you to extract the accounts payable (i.e. refund adjustments) in the following formats:

- Comma-Separated Values (CSV)
- Pipe-Separated Values (PSV)
- Tilde-Delimited File (TXT)
- Extensible Markup Language (XML)
- JavaScript Object Notation (JSON)

A new batch named **C1-APEXT** is introduced in this release. You need to set the **A/P Batch Code** field to **C1-APEXT** in the **Financial Transaction** tab of the **Installation Options** screen. On creating a refund adjustment, the system will then stamp the **C1-APEXT** batch control and batch run number whenever records are added in the **CI_ADJ_APREQ** table.

The **C1-APEXT** batch enables you to extract the accounts payable (i.e. refund adjustments) in a flat file. It considers all the adjustments created using the adjustment types where the **A/P Request Type Code** field is set to **REFUND** and whose corresponding record in the **CI_ADJ_APREQ** table has the latest batch run number. It then extracts the data in a flat file in the given format.

This batch is a multi-threaded batch. The multi-threading is based on adjustment ID and chunks for multi-threading are created based on numerical distribution of adjustment ID. It contains the following parameters:

Parameter Name	Mandatory (Yes or No)	Description
DIST-THD-POOL	No	Used to specify the thread pool on which you want to execute the batch.
File Name	Yes	Used to specify the name which you want to use for the flat file. Note: By default, the file name is set to DummyRecords .
File Path	Yes	Used to specify the directory where you want to store the extracted 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: <ul style="list-style-type: none"> • @SHARED_DIR – Used when you want to store the extracted file in the shared directory on the server. For example, you can set the value to @SHARED_DIR/output .

Parameter Name	Mandatory (Yes or No)	Description
File Format	Yes	Used to specify the format in which you want to extract the data. The valid values are: <ul style="list-style-type: none"> • CSV – Used when you want to extract the data in the Comma-Separated Values (CSV) format. • PSV - Used when you want to extract the data in the Pipe-Separated Values (PSV) format. • TILD - Used when you want to extract the data in the Tilde-Delimited File (TXT) format. • XML - Used when you want to extract the data in the Extensible Markup Language (XML) format. • JSON - Used when you want to extract the data in the JavaScript Object Notation (JSON) format.
Header Date Format	Yes	Used to specify the format in which the date should printed in the header of the extracted file. The header date indicates when the flat file was extracted in ORMB. <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>Note: By default, the date format is set to dd-MM-yyyy.</p> </div>

Once the data is extracted in the given format at the specified location, the status of the accounts payable request in the **CI_ADJ_APREQ** table is changed from **N** (i.e. Not Ready) to **R** (i.e. Ready for Payment). You can then share the flat file with an external system (i.e. Accounts Payable Processing System). The external system will then accordingly make the refund payments to the customers as appropriate.

Updating Accounts Payable Request in ORMB

Once the external system (i.e. Accounts Payable Processing System) makes a payment against the accounts payable request or rejects the accounts payable request, ideally, the ORMB application should be accordingly updated to track the accounts payable request. Oracle Revenue Management and Billing provides the ability to receive the response for each accounts payable request sent to an external system. The system can receive the response in the following formats:

- Comma-Separated Values (CSV)
- Pipe-Separated Values (PSV)
- Tilde-Delimited File (TXT)
- Extensible Markup Language (XML)
- JavaScript Object Notation (JSON)

You can send response for multiple accounts payable requests in a single file. While sending the response, you need to ensure that the file contains the following for each accounts payable request:

- **Payment Selection Status** – Used to indicate the status of accounts payable request. The valid values are:
 - **P** – Used when the refund is paid to the customer.
 - **X** – Used when the refund is rejected in the external system.
- **Accounts Payable Request ID** – Used to identify the accounts payable request in ORMB.
- **Cancel Reason** – Used to specify the adjustment cancel reason. It is required when the payment selection status is set to **X**. Here, you can specify an adjustment cancel reason which is already defined in the system.
- **Payment Method** – Used to indicate how the payment is made to the customer. It is required when the payment selection status is set to **P**. Here, you can specify a value which is already defined in the **PYMNT_METHOD_FLG** lookup field.
- **Payment Currency** – Used to specify the currency in which the payment is made. It is required when the payment selection status is set to **P**. Here, you can specify a currency which is already defined in the system.
- **Payment Date** – Used to specify the date when the payment is made. It is required when the payment selection status is set to **P**. Here, you can specify the date in the **YYYY-MM-DD** format.
- **Payment Amount** – Used to specify the payment amount. It is required when the payment selection status is set to **P**.
- **Payment Number** – Used to specify the payment number using which you can track the payment in the external system. It is required when the payment selection status is set to **P**.

Oracle Revenue Management and Billing enables you to upload and process the response through the **File Upload Interface** utility. We have shipped the following three file request types with the product:

- **APUPDATECSV** – Used when you want to upload and process the accounts payable response in the **Comma-Separated Values (CSV)** format.
- **APUPDATEXML** - Used when you want to upload and process the accounts payable response in the **Extensible Markup Language (XML)** format.
- **APUPDATEJSON** - Used when you want to upload and process the accounts payable response in the **JavaScript Object Notation (JSON)** format.

You can create similar file request types for the **Pipe-Separated Values (PSV)** and **Tilde-Delimited File (TXT)** formats. Once you upload and process the files through the **File Upload Interface** utility, the system will update the following columns in the respective table depending on the response:

Accounts Payable Request is....	Table Name	Column Name
Accepted (i.e. Refund is paid)	CI_ADJ_APREQ	PYMNT_SEL_STAT_FLG
		PYMNT_METHOD_FLG
		CURRENCY_PYMNT
		PYMNT_DT
		PAID_AMT

Accounts Payable Request is....	Table Name	Column Name
		PYMNT_ID
Rejected (i.e. Refund is canceled)	CI_ADJ_APREQ	PYMNT_SEL_STAT_FLG
	CI_ADJ	CAN_RSN_CD

Delinquency Process Type Preference

A delinquency process type preference enables you to set the attributes which are used while creating a delinquency process for a person or an account. You need to create two distinct delinquency process type preferences – one for the fully insured group business and another for the fully insured individual business. A delinquency process type preference for the fully insured group business enables the system to create a delinquency process for a person or an account to which the policies are billed. However, a delinquency process type preference for the fully insured individual business enables the system to create a delinquency process for an account to which the individual memberships are billed. Once you create a delinquency process type preference for the fully insured group business, you need to specify the preference code in the **Delinquency Process Type Field Mapping – Group** parameter of the **C1-DLNQMNRL** algorithm. Similarly, Once you create a delinquency process type preference for the fully insured individual business, you need to specify the preference code in the **Delinquency Process Type Field Mapping – Individual** parameter of the **C1-DLNQMNRL** algorithm.

You can define, edit, delete, and copy a delinquency process type preference using the **C1-FieldMapping** business object through the **Field Mapping** screen. You can create a delinquency process type preference using the **Delinquency Process Type** preference category. While creating a delinquency process type preference for the fully insured individual business, you can specify the following attributes:

- **Active Individual Delinquency** – Used to indicate the delinquency process type using which you want to create a delinquency process for an account when its main customer is an active customer. An individual is deemed to be an active customer when at least one membership of the individual customer is in the **Active** status.
- **Effectuated Individual Delinquency for Ancillary** – Used to indicate the delinquency process type using which you want to create a delinquency process for an account when the following conditions are met:
 - The **Active Individual Delinquency** attribute is not defined in the delinquency process type preference
 - Paid through date is present on the account
 - Earliest unpaid coverage period does not include subsidy charges
 - Individual is enrolled to a health plan which belongs to a non-medical health product
- **Effectuated Individual without Benefit** – Used to indicate the delinquency process type using which you want to create a delinquency process for an account when the following conditions are met:
 - The **Active Individual Delinquency** attribute is not defined in the delinquency process type preference
 - Paid through date is present on the account

- Earliest unpaid coverage period does not include subsidy charges
- Individual is enrolled to a health plan which belongs to a medical health product
- **Effectuated Individual with Benefit** – Used to indicate the delinquency process type using which you want to create a delinquency process for an account when the following conditions are met:
 - The **Active Individual Delinquency** attribute is not defined in the delinquency process type preference
 - Paid through date is present on the account
 - Earliest unpaid coverage period includes subsidy charges
- **Benefit Sub Type** – Used to indicate the benefit sub type. The system considers the benefit (i.e. subsidy) charges of the specified benefit sub type to derive the appropriate delinquency process type for an account. The system considers this attribute when the following conditions are met:
 - The **Active Individual Delinquency** attribute is not defined in the delinquency process type preference
 - Paid through date is present on the account
 - The value is defined for the **Effectuated Individual with Benefit** attribute in the delinquency process type preference
- **Non-Effectuated Individual Delinquency** – Used to indicate the delinquency process type using which you want to create a delinquency process for an account when the following conditions are met:
 - The **Active Individual Delinquency** attribute is not defined in the delinquency process type preference
 - Paid through date is not present on the account
- **Termed Individual Delinquency** – Used to indicate the delinquency process type using which you want to create a delinquency process for an account when its main customer is an inactive customer. An individual is deemed to be an inactive customer when all memberships of the individual customer are in the **Terminated** status.

Points to Note:

All the above-mentioned attributes are not applicable for the fully insured group business.

The **Active Individual Delinquency** and **Termed Individual Delinquency** attributes are required. If you do not specify the **Active Individual Delinquency** attribute, you need to specify the following attributes – **Effectuated Individual Delinquency for Ancillary, Effectuated Individual without Benefit, Effectuated Individual with Benefit, Benefit Sub Type, and Non-Effectuated Individual Delinquency**. However, if you specify all these attributes including the **Active Individual Delinquency** attribute, the system will ignore the other attributes and only give priority to the **active Individual Delinquency** attribute.

However, while creating a delinquency process type preference for the fully insured group business, you can specify the following attributes:

- **Active Group Customers – Account Level** – Used to indicate the delinquency process type using which you want to create the delinquency process for an account when its main customer is an active customer. A person is deemed to be an active customer when at least one policy associated with the person (i.e. policy holder) is in the **Active** status.
- **Active Group Customers – Person Level** – Used to indicate the delinquency process type using which you want to create the delinquency process for an active person. A person is deemed to be an active customer when at least one policy associated with the person (i.e. policy holder) is in the **Active** status.
- **Inactive Group Customers – Account Level** – Used to indicate the delinquency process type using which you want to create the delinquency process for an account when its main customer is an inactive customer. A person is deemed to be an inactive customer when all policies associated with the person (i.e. policy holder) are in the **Terminated** status.
- **Inactive Group Customers – Person Level** – Used to indicate the delinquency process type using which you want to create the delinquency process for an inactive person. A person is deemed to be an inactive customer when all policies associated with the person (i.e. policy holder) are in the **Terminated** status.

Points to Note:

All the above-mentioned attributes are not applicable for the fully insured individual business.

All the above attributes are mandatory while creating a delinquency process type preference for the fully insured group business.

Delinquency Process Preference

A delinquency process preference enables you to set the attributes which are used at various stages in the lifecycle of a delinquency process. In case you offer the **Delinquency Management** feature to both the fully insured group and individual customers, the system will use the same delinquency process preference for both the fully insured group business and the fully insured individual business. Some attributes in the delinquency process preference are common for both the fully insured group business and the fully insured individual business, while some attributes are specific to the fully insured group business or the fully insured individual business. The system considers the delinquency process preference which is specified in the **Delinquency Process Field Mapping** option type of the **DELINPROC** feature configuration.

You can define, edit, delete, and copy a delinquency process preference using the **C1-FieldMapping** business object through the **Field Mapping** screen. You can create a delinquency process preference using the **Delinquency Process** preference category. While creating a delinquency process preference, you need to specify the following attributes:

- **Advance Deposit Account Relationship Type** – Used to specify the account relationship type. It is used to identify the advance deposit account of a parent customer (i.e. policy holder) when the advance deposit level is set to **PG** or **POLI**, respectively. This attribute is required when the group customer wants to allocate advance deposit to settle its overdue bills. This attribute is only applicable for the fully insured group business.
- **Advance Deposit Contract Type** – Used to specify the advance deposit contract type. It is used to identify the advance deposit contract on the parent customer’s advance deposit account or linked to the policy when the advance deposit level is set to **PG** or **POLI**, respectively. This attribute is required when the group customer wants to allocate advance deposit to settle its overdue bills. This attribute is only applicable for the fully insured group business.
- **Awaiting Membership Cancellation Reason** – Used to indicate the reason why you want to cancel an individual membership which is billed to an account. Here, you must specify a status reason which is already defined for the **Active** status of the **C1-IndMembership** business object.
- **Delinquency Process Approval In Progress Status** – Used to specify the status to which a delinquency process should be transitioned when approval process is configured for any manual action in a delinquency process type. This attribute is required only when you opt to configure approval process for any manual action in the delinquency process.
- **Delinquency Process Automatic Reinstatement Stopped** – Used to specify the status to which a delinquency process should be transitioned when automatic reinstatement of the delinquency process is stopped due to some reason.
- **Delinquency Process Canceled Status** – Used to specify the status to which a delinquency process should be transitioned when it is canceled due to any reason. It is also referred while resuming a delinquency process due to cancellation of a payment or an adjustment.
- **Delinquency Process Completed Status** – Used to specify the status to which a delinquency process should be transitioned when all its delinquency events are triggered successfully.
- **Delinquency Process Customer Reinstated Status** – Used to specify the status to which a delinquency process should be transitioned when the policies or individual memberships billed to the person or account are reinstated. It is also referred while resuming a delinquency process due to cancellation of a payment or an adjustment.
- **Delinquency Process Hold Status** – Used to specify the status to which a delinquency process should be transitioned when you manually hold the delinquency process or when the system automatically holds the delinquency process if the certain conditions are met.
- **Delinquency Process Initiate Cancel Status** – Used to specify the status to which a delinquency process should be transitioned when you manually cancel the delinquency process.
- **Delinquency Process Initiate Hold Status** – Used to specify the status to which a delinquency process should be transitioned when you manually hold the delinquency process.
- **Delinquency Process Initiate Reinstatement Status** - Used to specify the status to which a delinquency process should be transitioned when you manually reinstate the delinquency process.
- **Delinquency Process Initiate Release Status** - Used to specify the status to which a delinquency process should be transitioned when you manually release the hold on delinquency process.
- **Delinquency Process Initiated Status** - Used to specify the status to which a delinquency process should be transitioned when it is created for a person or an account.
- **Delinquency Process InProgress Status** – Used to specify the status to which a delinquency process should be transitioned when its first event is triggered.

- **Delinquency Process Pending Termination Status** – Used to specify the status to which a delinquency process should be transitioned on the termination request date when the **C1-DPMON** batch is executed.
- **Delinquency Process Reinstate Status** - Used to specify the status to which a delinquency process should be transitioned when you manually reinstate the delinquency process or when the system automatically reinstates the delinquency process if the certain conditions are met.
- **Delinquency Process Released Status** – Used to specify the status to which a delinquency process should be transitioned when you manually release the hold on the delinquency process or when the system automatically releases the hold on the delinquency process if the certain conditions are met.
- **Delinquency Process Terminated Status** – Used to specify the status to which a delinquency process should be transitioned when the policies or individual memberships billed to the respective account are terminated in ORMB. This attribute is referred while executing the **C1-DPMON** batch.
- **Delinquency Reinstatement Payment/Adjustment Canceled Status** – Used to specify the status to which a delinquency process should be transitioned when you want to resume the reinstated delinquency process due to cancellation of a payment or an adjustment.
- **Membership Termination Reason for Delinquency** – Used to indicate the reason why you want to terminate an active individual membership which is billed to an account. Here, you must specify a status reason which is already defined for the **Terminated** status of the **C1-IndMembership** business object.
- **Policy Characteristic Type for Customer Contact** – Used to indicate the characteristic type using which you want to store the policy ID on the customer contact. Here, you must specify a characteristic type where the characteristic entity is set to **Customer Contact**. This attribute is also referred while deriving the policy ID from the customer contact.
- **Policy Termination Reason for Delinquency Process** - Used to indicate the reason why you want to terminate an active policy which is billed to a person or an account. Here, you must specify a status reason which is already defined for the **Terminated** status of the **C1-POLICY** business object.

Routing Method – Contact Method Mapping Preference

A routing method – contact method mapping preference enables you to map a bill routing method to a preferred contact method. The system will then derive the preferred contact method for the bill routing method from the routing method – contact method mapping preference while creating a customer contact for a person or an account during the delinquency management.

In case you offer the **Delinquency Management** feature to both the fully insured group and individual customers, the system will use the same routing method – contact method mapping preference for both the fully insured group business and the fully insured individual business. The system considers the routing method – contact method mapping preference which is specified in the **Routing Method – Contact Method Mapping** option type of the **DELINPROC** feature configuration.

You can define, edit, delete, and copy a routing method – contact method mapping preference using the **C1_BillRouteCntMethMap** business object through the **Field Mapping** screen. While creating a routing method – contact method mapping preference, you can specify the following attributes:

- **Bill Routing Method** – Used to indicate the bill routing method for which you want to map the preferred contact method. Here, the list includes all the values which are defined in the **BILL_RTG_METH_FLG** lookup field.
- **Preferred Contact Method** – Used to indicate the preferred contact method for the bill routing method. Here, the list includes all the values which are defined in the **CONTACT_METH_FLG** lookup field.

Enhancements (Generic)

This section lists the enhancements made to the following features which can be used in both financial services and health insurance domains:

- [C1-FLUPLD Feature Configuration](#)
- [File Upload Interface](#)
- [FOP Reporting](#)

C1-FLUPLD Feature Configuration

The following option types are newly added to the **C1-FLUPLD** feature configuration:

- **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.
- **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.
- **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.

For more information about these option types, refer to the **Setting the C1-FLUPLD Feature Configuration** section in *Oracle Revenue Management and Billing Installation Guide*.

File Upload Interface

The following changes are made to the File Upload Interface feature:

- Now, you can upload and transform the Pretty Good Privacy (PGP) encrypted files through the **File Upload Interface** utility.
- Now, in the On Premise and Cloud Service application environments, the **@INSTALL_DIR** variable is no longer supported. You cannot upload or download the files from the installed directory on the server. You can only upload or download the files from the shared directory on the server using the **@SHARED_DIR** variable.
- Now, you can upload or download a file with a maximum size of 10 MB from the **File Management System** screen.

FOP Reporting

The following changes are made to the FOP Reporting feature:

- Now, in the On Premise and Cloud Service application environments, the **@INSTALL_DIR** variable is no longer supported. You cannot extract or read the files from the installed directory on the server. You can only extract or read the files from the shared directory on the server using the **@SHARED_DIR** variable.

- The following parameters are removed from the **Report Generation (C1-RPTGN)** batch:
 - Report Query
 - Report Query Parameter
- The following parameters are newly introduced in the **Report Generation (C1-RPTGN)** batch:
 - **Batch Report Zone** – Used to specify the zone using which you want to present the extracted data in the PDF or Excel format. For example, if you want to generate PDF report for each division, you can create a zone with the following query:

```
SELECT TRIM(CIS_DIVISION) AS CIS_DIVISION, :START_DATE AS START_DATE FROM
CI_CIS_DIVISION
```

Points to Note:

You can use a bind variable in the zone query. For example, :START_DATE as shown in the above sample query. You can specify multiple bind variables, if required.

The bind variables must be alphanumeric and prefixed with the colon (:). The values for these bind variables are specified while executing the **C1-RPTGN** batch.

- **Binding Parameter Names used in the Zone** – Used to specify the bind variables that are used in the zone query. You can specify multiple bind variables separated by the **Tilde (~)** symbol in this parameter.
- **Binding Parameter Values for the above Names** – Used to specify the values for the bind variables used in the zone query. You can specify multiple values separated by the **Tilde (~)** symbol in this parameter.

Enhancements (Specific to Financial Services)

This section lists the enhancements made to the following features which can be used in the financial services domain:

- [Billing Anomaly Detection with Artificial Intelligence \(AI\) and Machine Learning \(ML\)](#)
- [Statements in TWIST and CAMT Formats](#)

Billing Anomaly Detection with Artificial Intelligence (AI) and Machine Learning (ML)

The following changes are made to the Billing Anomaly Detection feature:

- The **Trial Bill** field is removed from the **Search Anomaly** zone of the **Diagnostic Central** screen. This is because we do not support anomaly detection for the trial bills.
- Now, the system displays specific anomaly reasons in the **Anomaly Reason** list. When you search for anomalies using the **Account/Bill Level Anomaly** option from the **Search By** list, the system lists the following options in the **Anomaly Reason** list:
 - **Amount Variation** – Used to search the bill anomalies that are detected due to variation in the bill amount when compared with the historical data.
 - **Price Item Added** - Used to search the bill anomalies that are detected due to an additional price item when compared with the historical data.
 - **Price Item Deleted** - Used to search the bill anomalies that are detected due to removal of a price item when compared with the historical data.
 - **Bill Cycle Code Change** - Used to search the bill anomalies that are detected due to change in the account's bill cycle when compared with the historical data.
 - **Currency Code Change** - Used to search the bill anomalies that are detected due to change in the account's currency when compared with the historical data.

In the **Search Results** section, the system displays the anomaly reasons in the following formats:

Anomaly Reason	Format	Example	Comments
Amount Variation	Bill Amount: {VERY HIGH or VERY LOW}:{Current Bill Amount}	Bill Amount: VERY HIGH:16503	Current bill amount is higher than the historical bills.
		Bill Amount: VERY LOW:26000	Current bill amount is lower than the historical bills.

Anomaly Reason	Format	Example	Comments
Price Item Added	Priceitem(s) Added: {Price Item Code},{Price Item Code}	Priceitem(s) Added: J0006,J0008	Charges of the new price items are added in the current bill when compared with the historical bills.
Price Item Deleted	Priceitem(s) Removed: {Price Item Code},{Price Item Code}	Priceitem(s) Removed: J0001,J0004	Charges of some price items are not included in the current bill when compared with the historical bills.
Bill Cycle Code Change	Bill Cycle Code Change: {Old Bill Cycle Code}:{New Bill Cycle Code}	Bill Cycle Code Change: VP01:VP02	Bill cycle of the account has changed before generating the current bill when compared with the historical bills.
Currency Code Change	Currency Code Change: {Old Currency Code}:{New Currency Code}	Currency Code Change: USD:EUR	The currency code of the account has changed before generating the current bill when compared with the historical bills.

However, when you search for anomalies using the **Bill Segment Level Anomaly** option from the **Search By** list, the system lists the following options in the **Anomaly Reason** list:

- **BSEG Amount Variation** – Used to search the bill segment anomalies that are detected due to variation in the bill segment amount when compared with the historical data.
- **SQL Variation** - Used to search the bill segment anomalies that are detected due to variation in the SQL amount when compared with the historical data.
- **Pricing Change** - Used to search the bill segment anomalies that are detected due to change in pricing when compared with the historical data.

In the **Search Results** section, the system displays the anomaly reasons in the following formats:

Anomaly Reason	Format	Example	Comments
BSEG Amount Variation	Bill Segment Amount: {VERY HIGH or VERY LOW}:{Current Bill Segment Amount}	Bill Segment Amount: VERY HIGH:401	Current bill segment amount is higher than the historical bill segments.
		Bill Segment Amount: VERY LOW:26000	Current bill segment amount is lower than the historical bill segments.

Anomaly Reason	Format	Example	Comments
SQI Variation	SQI Value: {SQI}:{VERY HIGH or VERY LOW}:{Current SQI Amount}	SQI Value: BK-NBR:VERY LOW:2600	SQI amount in the current bill segment is higher than SQI amount in the historical bill segments.
		SQI Value: BK-NBR:VERY HIGH:23000	SQI amount in the current bill segment is lower than SQI amount in the historical bill segments.
Pricing Change	Pricing Change: {Price Assignment ID}:{AGP or STD}	Pricing Change: 7030016157:AGP	Another pricing is used to charge a price item in the current bill segment when compared with the historical bill segments.
		Pricing Change: 4180016165:STD	
			<p>Points to Note:</p> <p>AGP stands for the agreed pricing and STD stands for the standard pricing.</p> <p>The system considers all the pricing as AGP when its assignment type in the CI PRICEASGN table is set to PRTY and all the pricing as STD when its assignment type in the CI PRICEASGN table is set to PLST.</p>

- The system enables you suppress some anomalies at the bill and bill segment levels. A new option type named **Suppression Flag** is introduced in the **C1-BRMLINOPS** feature configuration. It enables you to suppress certain anomalies at the bill and bill segment levels. The valid values for the **Suppression Flag** option type are **Y** and **N**. If you set the **Suppression Flag** option type to **Y**, the system will suppress the following:
 - The amount variation anomaly at the bill level when the currency code of the account is changed before generating the current bill. This is because it is obvious that the bill amount will change due to change in the account's currency and therefore the main reason for anomaly is Currency Code Change and not BSEG Amount Variation.
 - The pricing change anomaly at the bill segment level when the standard pricing is used to generate the bill segment.

- The pricing change and bill segment amount variation anomalies at the bill segment level when the standard pricing is used to generate the bill segment.

However, if you set the **Suppression Flag** option type to **Y**, the system will not suppress all other anomalies including the following:

- The pricing change, bill segment amount variation, and SQI amount variation anomalies at the bill segment level when the standard pricing is used to generate the bill segment.
- The pricing change anomaly at the bill segment level when the agreed pricing is used to generate the bill segment.
- The pricing change and bill segment amount variation anomalies at the bill segment level when the agreed pricing is used to generate the bill segment.
- The pricing change, bill segment amount variation, and SQI amount variation anomalies at the bill segment level when the agreed pricing is used to generate the bill segment.

If you set the **Suppression Flag** option type to **N**, the system will not suppress any anomalies at the bill or bill segment level. If you do not specify the value for this option type, the system, by default, sets it to **N**.

Statements in TWIST and CAMT Formats

The following changes are made to the TWIST 3.1 and camt.086.001.04 formats:

- We have added 10 string and 5 BigDecimal type user-defined fields for each value object (getter/setter method) to use the Optional and Conditional tags.

Enhancements (Specific to Insurance)

This section lists the enhancements made to the following features which can be used in the health insurance domain:

- [Inbound Message](#)
- [Offset Request](#)
- [Refund/Write Off Request](#)
- [Payment Request](#)
- [Customer 360° View](#)
- [Binder Payment Preference \(Field Mapping\)](#)
- [Health Product 360° View](#)
- [Reconciliation](#)
- [Membership](#)
- [Invoice Request](#)
- [Individual Health Insurance Billing](#)
- [Individual Health Insurance Pricing](#)
- [Field Mapping](#)
- [Business Rules](#)
- [Parameter](#)
- [Collection Class](#)
- [Customer Class](#)
- [Adjustments](#)
- [C1-MembershipRepricing Business Service](#)
- [Payments](#)
- [Individual Membership](#)
- [Subscription Tier Derivation for Membership](#)
- [Repricing](#)
- [C1-ASOBLNG Feature Configuration](#)
- [C1_CMO Feature Configuration](#)
- [C1-INVREQ Feature Configuration](#)
- [Membership Benefits](#)
- [Fully Insured Pricing](#)
- [Fully Insured Group Policy](#)

Inbound Message

The following changes are made to the Inbound Message feature:

- The **C1-HCInboundMessage** business object is enhanced to do the following:
 - Maintain skip auto maintenance information for an account
 - Maintain the drag days for an account
 - Maintain the collection and advance deposit details of a person
 - Maintain the status of an address for an entity
 - Maintain the sequence and repricing after date for a membership
 - Maintain the advance deposit details of a policy
- You can set the advance deposit level to **PG** (i.e. Parent Customer) or **POLI** (i.e. Policy). If the advance deposit level is set to **PG**, you need to specify the advance deposit amount and advance deposit grace days in the **personData** section. However, if the advance deposit level is set to **POLI**, you need to specify the advance deposit applicability, advance deposit amount, advance deposit grace days, advance deposit account identifier type, and advance deposit account identifier in the **policyData** section. Note that all advance deposit details are required in the **policyData** section if the advance deposit applicability is set to **Y**.
- The following tags are newly introduced in the health care inbound message schema:

Parent Tag	New Tags
<accountData>	<dragDays></dragDays> <skipAutoMaintenanceSw></skipAutoMaintenanceSw>
<personData>	<collectionClass></collectionClass> <postponeCreditReviewUntil></postponeCreditReviewUntil> <lastCreditReviewDate></lastCreditReviewDate> <dragDays></dragDays> <advanceDepositAmount></advanceDepositAmount> <advanceDepositGraceDays></advanceDepositGraceDays> <advanceDepositLevel></advanceDepositLevel>
<address>	<addressStatus></addressStatus>
<memberData>	<sequence></sequence> <repriceAfterDate></repriceAfterDate>
<personAddressOverride>	<addressStatus></addressStatus>
<policyData>	<advanceDepositApplSW></advanceDepositApplSW> <advanceDepositAmount></advanceDepositAmount> <advanceDepositGraceDays></advanceDepositGraceDays>

Parent Tag	New Tags
	<pre><advanceDepositAccountIdentifierType></advanceDepositAccountIdentifierType> <advanceDepositAccountIdentifierValue></advanceDepositAccountIdentifierValue></pre>

- The following tag is renamed in the health care inbound message schema:

Parent Tag	Old Tag	New Tag
<personAddressOverride>	<postal>	<zip>

- The **C1-HCProdHlthPlnInbound** business object is enhanced to maintain the Health Insurance Oversight System (HIOS) ID for a health plan. The system allows you to specify the same HIOS ID for multiple health plans provided their coverage period is different and not overlapping.
- The following tag is newly introduced in the health product and plan inbound message schema:

Parent Tag	New Tags
<healthPlanData>	<hiosId></hiosId>

Offset Request

The following changes are made to the Offset Request feature:

- Until now, the **Offset Request** feature only supported offset between credit and debit bills. Now, to move the funds from the advance deposit contract to the unpaid bills through a delinquency process, the **Offset Request** feature is enhanced to support offset between credit adjustment and debit bills. The credit adjustment level offset is same as the bill segment level offset.

You can offset credit adjustment against the debit bills using an offset request type where the offset category is set to **Advance Deposit**. The advance deposit offset request is automatically created through an advance deposit delinquency process. The advance deposit offset request supports creating offset request adjustments with or without transfer adjustments.

- A new field named **Offset Category** is introduced in the **C1-OffsetRequestType** business object. The offset requests are now classified under the following two categories:
 - Regular** – The regular or traditional offset requests are manual offset requests. You can create a manual offset request using an offset request type where the offset category is set to **Regular**.
 - Auto Maintenance** – The auto maintenance offset requests are automatic offset requests. You can create an auto maintenance offset request using an offset request type where the offset category is set to **Auto Maintenance**.
 - Advance Deposit** - The advance deposit offset requests are automatic offset requests. You can create an advance deposit offset request using an offset request type where the offset category is set to **Advance Deposit**.

- The **Offset Request** screen is enhanced to support an auto maintenance offset request. A new zone named **Auto Maintenance Financial Transactions** appears in the **Main** tab of the **Offset Request** screen. It lists the financial transactions of the account for which the corresponding offset request adjustments are created during the auto maintenance process.
- While viewing the details of an auto maintenance offset request, the system displays the **Auto Maintenance Financial Transactions** zone instead of the **Selected Bills** and **Offset Request Adjustments** zones in the **Main** tab of the **Offset Request** screen. The system also displays the account information in the **Main** section of the **Offset Request** zone.
- If you set the **Skip Auto Maintenance for Accounts with Open Offset Request (Y/N)** parameter to **N** while executing the **C1-AUTOM** batch, the system will create an auto maintenance offset request for an account even if the regular offset request exists for the account in the non-final status. The offset request feature is enhanced to handle such scenarios.

The system will throw an appropriate error message when you process the manual offset request containing bills or bill line items which are closed during the auto maintenance of the account.

- The **Offset Request** screen is enhanced to support the advance deposit offset requests. A new zone named **Adjustment** appears in the **Main** tab of the **Offset Request** screen. It lists the credit adjustment of the advance deposit contract which is used for offsetting the overdue bills through a delinquency process.

Refund/Write Off Request

The following changes are made to the Refund/Write Off Request feature:

- If you set the **Skip Auto Maintenance for Accounts with Open Refund or Write Off Request (Y/N)** parameter to **N** while executing the **C1-AUTOM** batch, the system will create an auto maintenance offset request for an account even if the refund or write off request exists for the account in the non-final status. The refund/write off request feature is enhanced to handle such scenarios.

The system will throw an appropriate error message when you process the refund/write off request containing bills or bill line items which are closed during the auto maintenance of the account.

Payment Request

The following changes are made to the Payment Request feature:

- If you set the **Skip Auto Maintenance for Accounts with Open Payment, Payment Transfer Request (Y/N)** parameter to **N** while executing the **C1-AUTOM** batch, the system will create an auto maintenance offset request for an account even if the payment creation or payment transfer request exists for the account in the non-final status. The payment request feature is enhanced to handle such scenarios.

The system will apply the payment on an excess credit contract of the account when you process the payment creation or payment transfer request containing bills or bill line items which are closed during the auto maintenance of the account.

Customer 360° View

The following changes are made to the Customer 360° View feature:

- Now, you can view the skip auto maintenance information of the account in the **Account Information** zone of the **Account** tab. Note that the **Skip Auto Maintenance** field appears for the account only when it is set to **Y**.
- A new section named **Person Collection** is introduced in the **Person Information** zone of the **Person** tab. It displays the person's collection and advance deposit details, such as collection class, last credit review date, drag days, advance deposit amount, advance deposit grace days, and advance deposit level. Note that the fields appear in the **Person Collection** section only when the data is available. In addition, note that the **Advance Deposit Amount** and **Advance Deposit Grace Days** fields appear in the **Person Collection** section only when the advance deposit level for a parent customer is set to **Parent Customer**.
- If the advance deposit level for a parent customer is set to **Policy**, the advance deposit details, such as advance deposit applicability, advance deposit amount, advance deposit grace days, and advance deposit account are displayed in the **Main** section of the **Policy Information** zone on the **Policy** tab. Note that the **Advance Deposit Amount**, **Advance Deposit Grace Days**, and **Advance Deposit Account** fields appear only when the **Advance Deposit Applicability** option is selected.
- You can now view the information of a delinquency process when the delinquency process is created for an overdue bill of the account. A new column named **Delinquency Process Information** is introduced in the **Open Bills** zone of the **Account** tab.
- Now, while viewing the pricing group details of a self-funded pricing rule, you can view the number of rules defined in the respective pricing group. A new column named **Rules** is added in the **Bill Group Pricing Groups** zone of the **Pricing Information** tab.

Binder Payment Preference (Field Mapping)

The following attributes are newly introduced for the **Binder Payment** preference category:

- **Membership Status Reason when Binder Payment Cancelled** – Used to specify the membership status reason code that you want to use while creating a To Do notification during the binder payment cancellation.
- **To Do Type for Binder Payment Cancelled** – Used to specify the To Do type using which you want to create a To Do during the binder payment cancellation. Note that the **C1-BPCNC** To Do type is shipped with the product.

Health Product 360° View

The following changes are made to the Health Product 360° View feature:

- The **C1-HealthPlan** business object is enhanced to maintain the HIOS ID for a health plan. The system allows you to specify the same HIOS ID for multiple health plans provided their coverage period is different and not overlapping.

- Now, you can search for a health plan using its HIOS ID. A new field named **HIOS ID** is introduced in the **360° Search** zone when you search using the **Health Plan Details** option from the **Search By** list.
- You can view the HIOS ID of the health plan in the **Plan Information** zone of the **Health Plan 360° Information** screen. Note that this field appears only when the HIOS ID is available for the health plan.

Reconciliation

The following changes are made to the Reconciliation feature:

- Now, the Reconciliation feature is enhanced to support the fully insured individual business.
- The **C1-PayInstruction** business object is enhanced to store the following information in a pay instruction for the fully insured individual business:
 - Health Plan Code
 - HIOS ID
 - Payor Identifier Type
 - Payor Identifier Value
 - Payment Type
 - Price Item
- The **Health Plan** and **Payment Type** fields are added in the **Pay Instructions** zone of the **Reconciliation** screen. These fields appear only when you are viewing the details of a reconciliation object which is created for the fully insured individual business. However, the **Policy Number** field appears in the **Pay Instructions** zone when you are viewing the details of a reconciliation object which is created for the fully insured group business.
- The **Payor Identifier Type**, **Payor Identifier**, **Health Plan**, **HIOS ID**, and **Payment Type** columns are added in the **Pay Instructions** zone of the **Reconciliation** screen. These columns appear only when you are viewing the details of a reconciliation object which is created for the fully insured individual business. However, the **Policy Number** and **Plan Number** columns appear in the **Pay Instructions** zone when you are viewing the details of a reconciliation object which is created for the fully insured group business.
- The **Health Plan**, **HIOS ID**, **Payment Type**, **Payor Identifier Type**, and **Payor Identifier** fields are added in the **Main** section of the **Pay Instruction** zone on the **Pay Instruction** screen. These fields appear only when you are viewing the details of a pay instruction which is created for the fully insured individual business. However, the **Policy Number** and **Plan Number** fields appear in the **Pay Instruction** zone when you are viewing the details of a pay instruction which is created for the fully insured group business.
- The **Health Plan**, **HIOS ID**, **Payment Type**, **Payor Identifier Type**, and **Payor Identifier** columns are added in the **Discrepancy Line Items** zone of the **Discrepancy Report** screen. These columns appear when you are viewing the discrepancy report for the fully insured individual business. However, the **Policy Number** and **Plan Number** columns appear when you are viewing the discrepancy report for the fully insured group business.

- Earlier, when a pay instruction was canceled, the system did not check whether there is any related pay instruction for it due to any take back or partial payment scenarios. For example, in the month of Jan 2023, a pay instruction (i.e. P1) of \$300 was received for the A1 account which was reconciled against its B1 (\$100) and B2 (\$100) line items. And, in the month of Feb 2023, a negative pay instruction (i.e. P2) of -\$100 received for the A1 account was reconciled against its Jan month's open pay instruction (i.e. P1). If, due to any reason, the P2 pay instruction was canceled, the system did not automatically reopen the P1 pay instruction and vice-versa. From this release onwards, this cancellation issue is handled for the take back and partial payment scenarios. Now, while canceling a pay instruction, the system checks whether there is any related pay instruction for it. If so, the system first accordingly reopens the related pay instruction and then cancels the primary pay instruction.

Membership (Screen)

The following changes are made to the Membership screen:

- You can now search for a health plan using its HIOS ID. A new field named **HIOS ID** is added in the **Health Plan Search** window.
- Appropriate log entries are added in the **Log** tab when an invoice request is created for the individual membership's account.
- While viewing the details of an individual membership, the **Reprice After Date** field appears in the **Main** section of the **Membership** zone.

Invoice Request

The following changes are made to the Invoice Request feature:

- This feature is enhanced to support the fully insured individual business along with other lines of business.
- A new value named **C1IM** is added in the **EXT_REF_TYPE_FLG** lookup field. Therefore, now, while searching for an invoice request, the following two values appear in the **External Reference Type** list:
 - Policy
 - Membership
- While creating an invoice request for an individual membership's account, the system considers the individual membership as the external reference and sets the external reference type of the invoice request to **Membership**.
- While viewing the details of an invoice request which is created for the fully insured individual business, the system does the following:
 - Displays the individual membership details in the **External Reference** section of the **Invoice Request** zone
 - Hides the **Account Information** column in the **Invoice Request Details** zone

However, while viewing the details of an invoice request which is created for the fully insured group business, the system does the following:

- Displays the policy details in the **External Reference** section of the **Invoice Request** zone

- Shows the **Account Information** column in the **Invoice Request Details** zone
- You can use the external reference such as policy ID or individual membership ID while searching for the invoice requests which are created for the fully insured group and fully insured individual businesses, respectively.

Individual Health Insurance Billing

The following changes are made to the Individual Health Insurance Billing feature:

- Until now, the system enabled you to create multiple accounts with a unique invoice type (i.e. account type) for a member who is financially responsible for an individual membership. Now onwards, you can create multiple accounts with the same invoice type (i.e. account type) for a member who is financially responsible for an individual membership.
- The charges of an individual membership are billed using the invoice type priority specifications given for the respective price item in the respective pricing rule type. Until now, as you were able to create only one account of a particular invoice type (i.e. account type), the system was able to derive the account of the member who is financially responsible for an individual membership using the given invoice type priority. But now, as the system enables you to create multiple accounts with the same invoice type, you need to specify the required account to which the individual membership should be billed as characteristic on the individual membership.

While fetching the characteristics on the individual membership, the system considers the characteristic types which are specified in the **Account Identifier Type Char Type** and **Account Identifier Value Char Type** option types of the **C1-ASOBLNG** feature configuration. Therefore, you need to define the account details for the individual membership using the characteristic types which are specified in the **Account Identifier Type Char Type** and **Account Identifier Value Char Type** option types through a health care inbound message.

Individual Health Insurance Pricing

The following changes are made to the Individual Health Insurance Pricing feature:

- Now, the system enables you to define separate maximum number of dependents for a minor and non-minor main subscriber. A new field named **Minor Maximum Number of Dependents** is available while defining a fully insured pricing business rule for the fully insured individual business. If the main subscriber is a non-minor, the system will use the maximum number of dependents specified in the **Maximum Dependents** column of the business rule. However, if the main subscriber is a minor, the system will use the maximum number of dependents specified in the **Minor Maximum Number of Dependents** column of the business rule.

Now, the following algorithms will recalculate the age of main subscriber and will use the maximum number of dependents from the **Minor Maximum Number of Dependents** or **Maximum Dependents** field, respectively, depending on whether the main subscriber is a minor or non-minor:

- C1_AGEDELG
- C1-DERSUBTR
- C1-MEMRLAUD

- The **No Of Days** parameter is removed from the **C1-PPRAMAGE** algorithm type. Instead, a new field named **Age Recalculation Number of Days** is available in the **Age Based Specific Additional Data** section when you define an age based pricing rule type using the **C1-PrcRuleTypeAgeBased** business object. You can define the number of days in the **Age Recalculation Number of Days** field.

Field Mapping

The following changes are made to the Field Mapping feature:

- The following new preference categories are introduced in this release:
 - Delinquency Process
 - Delinquency Process Type
 - Membership Repricing Reasons
 - Routing Method – Contact Method Mapping

Business Rules

The following changes are made to the Business Rules feature:

- The following new business rule categories are introduced in this release:
 - Delinquency Event Attributes Business Rule
 - Delinquency Grace Period Business Rule
 - Delinquency Miscellaneous Options Business Rule
 - Delinquency Termination Date Rule Business Rule
- Earlier, the **Business Rule Type** zone of the **Business Rule Type** screen was not operational. Now, while clicking the **Broadcast** icon corresponding to a business rule type in the **Search Business Rule Type** zone, the details of the business rule type appear in the **Business Rule Type** zone.
- The **Business Rule Type** screen enables you to maintain business rule criteria for the newly introduced business rules categories (listed above).

Parameter

The following changes are made to the Parameter feature:

- The following new source entities are introduced in this release:
 - Customer Class
 - Delinquency Event Type
 - Division
- The system enables you to define parameters which can be used while defining or editing a business rule criteria. A new option named **Business Rule Eligibility Criteria** is added to the **Parameter Usage** section. It appears only when the source entity is set to either of the following:
 - Account
 - Customer Class

- Delinquency Event Type
- Division
- Health Plan
- Health Product
- Membership
- Membership Person
- Policy
- Policy Plan

Collection Class

The following changes are made to the Collection Class feature:

- The following collection methods are introduced in this release:
 - **Self-Control Delinquency** – Used when you want to create a separate delinquency process for the accounts which belong to the collection class. These accounts will not be considered during the delinquency management of the parent person.
 - **Parental Delinquency** – This option is not applicable for the fully insured individual business.

Customer Class

The following changes are made to the Customer Class feature:

- Two new algorithms named **C1-CDPPAYCAN** and **C1-RODPDUEPX** are introduced in this release. They should be attached to the **Payment Cancellation** system event of a customer class to resume the delinquency process for the respective accounts on the payment cancellation. For more information about the algorithms, refer to the [Resuming a Delinquency Process on Payment Cancellation](#) section.

Adjustments

The following changes are made to the Adjustments feature:

- Two new algorithms named **C1-DPADJCAN** and **C1-RODPONADX** are introduced in this release. They should be attached to the **Adjustment Cancellation** system event of an adjustment type to resume the delinquency process for the account whose corresponding adjustment is canceled. For more information about the algorithms, refer to the [Resuming a Delinquency Process on Adjustment Cancellation](#) section.
- Two new fields named **Linked Entity Type** and **Linked Entity ID** appears in the **Main** tab of the **Adjustment** screen when you are viewing the details of an underpayment or short payment adjustment created for a bill. Note that the underpayment or short payment adjustment for a bill is created against a payment. Therefore, while viewing the details of an underpayment or short payment adjustment created for a bill, the **Linked Entity Type** field is set to **Payment** and the **Linked Entity ID** field is set to the respective payment ID.

C1-MembershipRepricing Business Service

The following changes are made to the **C1-MembershipRepricing** business service:

- The **C1-MembershipRepricing** business service is enhanced to do the following:
 - Maintain the collection and advance deposit details of a person
 - Maintain the status of an address for an entity
 - Maintain the repricing after date for a membership
- The following tags are newly introduced in the **C1-MembershipRepricing** business service schema:

Parent Tag	New Tags
<personData>	<collectionClass></collectionClass> <postponeCreditReviewUntil></postponeCreditReviewUntil> <lastCreditReviewDate></lastCreditReviewDate> <dragDays></dragDays> <advanceDepositAmount></advanceDepositAmount> <advanceDepositGraceDays></advanceDepositGraceDays> <advanceDepositLevel></advanceDepositLevel>
<address>	<addressStatus></addressStatus>
<membership>	<repriceAfterDate></repriceAfterDate>

Payments

The following changes are made to the Payments module:

- The length of the **CHECK_NBR** field in the following tables is increased from 10 to 30:
 - C1_PAY_TNDR_REQ
 - CI_PAY_TNDR
 - CI_PAY_TNDR_ST
 - CI_PEVT_DTL_ST
 - CI_UPLPAY_STG
- While uploading a payment data file through the **Payment Upload** feature, you can specify a check number up to 30 characters in the CSV file. Similarly, while creating a payment event or payment through either of the following screens, you can specify a check number up to 30 characters:
 - Payment Event Add
 - Payment Request
 - Payment Event Quick Add
 - Payment Quick Add
 - Payment Event Upload Staging

- Payment Upload Staging
- The following screens displays a check number up to 30 characters (if specified) while creating a payment:

Screen Name	Tab Name
Payment Event	Tenders
Payment Request	Main

- While searching a payment event or payment through either of the following screens, you can specify a check number up to 30 characters:
 - Payment Event Upload Staging Search
 - Payment Upload Staging Search
 - Payment Event Summary
- The **Payor Account Information** field appears in the **Payment Request** screen when you transfer a payment from the **Payment Event Summary** screen.
- A new field named **External Reference ID** appears when you search for a payment event using the **Tender Details** option from the **Search By** list. It enables you to search for the payment events whose tenders have a particular external reference ID.
- A new grid named **Payment Related Adjustments** appears in the **Main** tab of the **Payment** screen when the underpayment or short payment adjustments are directly linked to the payment and not to its payment segment.

Individual Membership

The following changes are made to the Individual Membership feature:

- Until now, while terminating an individual membership through a health care inbound message or through the membership repricing inbound web service, the system automatically set the status of all active dependent persons to **Inactive** after terminating the individual membership. The system never verified the dependent person's start date before inactivating the dependent person of an individual membership. Now, while inactivating a dependent person in the online or deferred (i.e. through the **C1-MEPRC** batch) mode during individual membership termination, the system checks whether the dependent person's start date is later than main subscriber's end date.

If the dependent person's start date is equal to or earlier than main subscriber's end date, the system will automatically set the status of the dependent person to **Inactive** after terminating the individual membership. However, if the dependent person's start date is later than main subscriber's end date, the system checks whether the dependent person's information is received along with the main subscriber details in the health care inbound message or in the membership repricing request.

If so, the system checks whether the status of the dependent person is set to **CNCL** (i.e. Canceled) in the health care inbound message or in the membership repricing request. If so, the system changes the status of the dependent person to **Canceled**. However, if the status of the dependent person is not set to **CNCL** (i.e. Canceled), the system throws an appropriate error message while processing the health care inbound message or the membership repricing request.

If the dependent person's information is not received in the health care inbound message or in the membership repricing request, the system sets the status of the dependent person (whose start date is later than the main subscriber's end date) to **Canceled**.

- Earlier, if a health care inbound message or a membership repricing request was received to cancel an individual membership which was already canceled in ORMB, the system threw an error message indicating that the active individual membership does not exist in the system. Now, the system will ignore such individual membership cancellation requests if the individual membership is already canceled in ORMB.
- While canceling an individual membership through a health care inbound message or through the membership repricing inbound web service, you must only send the main subscriber details (and not the dependent persons' details) in the health care inbound message or the membership repricing request where the main subscriber status is set to **CNCL** (i.e. Canceled). While processing the health care inbound message or the membership repricing request, the system will cancel the individual membership. Once the individual membership is canceled, the status of the active dependent persons is also changed to **CNCL** (i.e. Canceled). Note that the end date of the dependent persons is not updated when an individual membership is canceled.
- A new tag named **<sequence>** is introduced in the **memberData** section of the **C1-HCInboundMessage** business object schema. It helps to create or update the membership in the given sequence. It helps to address scenarios wherein multiple **memberData** sections with the same main subscriber needs to be processed in the correct order through the same health care inbound message.
- Now, the member ID should be unique for each individual membership on the health plan and all member persons in the membership should have the same member ID. The member ID field is not mandatory. But, if the member ID is specified for an individual membership, it should be unique at the membership level for a health plan.
- While updating an individual membership, the system derives the membership using the membership start date, membership end date, membership type (if given), member ID (if defined for the membership) and health plan combination.
- Earlier, while terminating an individual membership, the system used to set the end date of all active dependent persons to the termination date (i.e. to the main subscriber end date received in the inbound message). However, now, the system will check whether the end date of an active dependent person is earlier than the termination date. If so, the system will not update the end date of such dependent persons. The system will update the end date of only those active dependent persons whose end date is later than the termination date.

Subscription Tier Derivation for Membership

Until now, the system used to derive the subscription tier for a membership with the exact match (i.e. where the number of spouses, number of dependents, and number of young adults matches with the membership). If the exact match was not found for a membership, the system used to consider the subscription tier where the sum of self, number of spouses, number of dependents, and number of young adults exceeds the maximum member count specified in the subscription tier structure.

However, now, if the exact match is not found for a membership, the system will derive the nearest subscription tier for the membership. Let us understand this with the help of an example.

The following table illustrates the structure of a subscription tier named STEX1:

Subscription Tier	Self	Number of Spouses	Number of Dependents	Number of Young Adults
S	1	0	0	0
S1C	1	0	1	0
SC	1	0	3	0
SS	1	1	0	0
SSC	1	1	3	0
SCMax	1	0	99	0
SSMax	1	99	0	0
SYMax	1	0	0	99

Now, the system will derive the exact or nearest subscription tier for the memberships as shown in the below table:

Membership	Member Person	Main Subscriber	Relationship Type	Subscription Tier
M1	Mark	Yes	Self	S1C (Exact Match)
	Tom	No	Child	
M2	Roger	Yes	Self	SC (Nearest Match)
	Nancy	No	Child	
	James	No	Child	
M3	William	Yes	Self	SSMax (Nearest Match)
	Maria	No	Spouse	
	Jennifer	No	Spouse	
M4	Susan	Yes	Self	

Membership	Member Person	Main Subscriber	Relationship Type	Subscription Tier
	Jacob	No	Young Adult	SYMax (Nearest Match)

Repricing

The following changes are made to the Repricing feature:

- Until now, during the **C1-REPC1** batch execution, when a pricing rule was not derived for an individual membership, the system used to automatically delete the repricing entity detail record from the **CI_REPRC_ENTITY_DTL** table. Now, a new option type named **Delete Repricing** is introduced in the **C1-ASOBLNG** feature configuration. It enables you to configure whether the repricing entity detail record should be deleted or not when a pricing rule is not derived for an individual membership. You can set the option type to either of the following:
 - **Y** – Used when you want to delete the repricing entity detail record when a pricing rule is not derived for an individual membership.
 - **N** – Used when you do not want to delete the repricing entity detail record when a pricing rule is not derived for an individual membership. In such case, the system will change the status of the repricing entity detail record to **E** (i.e. Error).

C1-ASOBLNG Feature Configuration

The following option types are newly added to the **C1-ASOBLNG** feature configuration:

- **Additional Repricing** - Used to determine whether you want to create repricing entity detail records for all auditable elements (such as fields, business statuses, and characteristics) of the membership, member person, or person that are updated within the membership period. The valid values are:
 - **Y** – Used when you want to create repricing entity detail records for all auditable elements (such as fields, business statuses, and characteristics) of the membership, member person, or person that are updated within the membership period.
 - **N** – Used when you want to create repricing entity detail records in the traditional manner (i.e. one with the effective date as the membership start date and another with the effective date as the auditable element effective date) for each membership and pricing rule type combination.

Points to Note:

If you do not specify the value for this option type, the system, by default, sets it to N.

This option type is only applicable for the fully insured individual business.

For more information about this option type, refer to the [Additional Repricing Entity Detail Records](#) section.

- **Benefit Full Snapshot Upload** – Used to determine whether the full or partial snapshot of benefit charges is received for a group or individual membership through a health care inbound message or through a membership repricing request. The valid values are:
 - Y
 - N

Note: If you do not specify the value for this option type, the system, by default, sets it to **N**.

- **Delete Repricing** – Used to indicate whether the repricing entity detail record should be deleted when a pricing rule is not derived for an individual membership. The valid values are:
 - **Y** – Used when you want to delete the repricing entity detail record when a pricing rule is not derived for an individual membership.
 - **N** - Used when you do not want to delete the repricing entity detail record when a pricing rule is not derived for an individual membership. In such case, the system will change the status of the repricing entity detail record to **E** (i.e. Error).

Points to Note:

This option type is only applicable for the fully insured individual business.

This option type is optional. If you do not specify the value for this option type, the system, by default, deletes the repricing entity detail record when a pricing rule is not derived for an individual membership.

- **Individual Membership Configurations** – Used to indicate the preference which is created using the **Individual Membership** preference category.

Note: This option type is only applicable for the fully insured individual business.

- **Membership Active Status** – Used to specify the active status code of an individual membership. The system then considers the active status code while deriving the individual memberships using the **C1-MEMBDLQ** algorithm attached to the **Memberships Derivation** system event in the respective delinquency process type. Here, you must specify the status code which is defined in the lifecycle of the **C1-IndMembership** business object.

Note: This option type is only applicable for the fully insured individual business.

- **Membership Terminated Status** - Used to specify the terminated status code of an individual membership. The system then considers the terminated status code while deriving the individual memberships using the **C1-MEMBDLQ** algorithm attached to the **Memberships Derivation** system event in the respective delinquency process type. Here, you must specify the status code which is defined in the lifecycle of the **C1-IndMembership** business object.

Note: This option type is only applicable for the fully insured individual business.

- **Policy Active Status** - Used to specify the active status code of a fully insured group policy. The system then considers the active status code while deriving the policies using the **C1-POLDLQPA** algorithm attached to the **Policies Derivation** system event in the respective delinquency process type. Here, you must specify the status code which is defined in the lifecycle of the **C1-POLICY** business object.

Note: This option type is only applicable for the fully insured group business.

- **Zero Dollar Benefit Charge** – Used to determine whether the billable charge for zero benefit amount should be created or not. The valid values are:
 - Y
 - N

Note: If you do not specify the value for this option type, the system, by default, sets it to **N**.

C1_CMO Feature Configuration

The following option type is newly added to the **C1_CMO** feature configuration:

- Use Legacy Person Phone and Email

C1-INVREQ Feature Configuration

The following option type is newly added to the **C1-INVREQ** feature configuration:

- **Membership Search** – Used to specify the zone using which you want to search for an individual membership in the **Invoice Request** screen. The system then accordingly displays the **Search** window when you search for an invoice request using an external reference type named **Membership**.

Membership Benefits

The following changes are made to the Membership Benefits feature:

- A new option named **Premium Benefit** is introduced while defining and editing a benefit sub type. This option is applicable only for the fully insured group business. For more information about this option, refer to the [Prorate Membership Benefit and Corresponding Sponsor Charges](#) section.

Fully Insured Pricing

The following changes are made to the Fully Insured Pricing feature:

- A new column named **Proration Required** is introduced in the **Price Items** section while defining or editing a benefit pricing rule type. For more information about the **Proration Required** option, refer to the [Prorate Membership Benefit and Corresponding Sponsor Charges](#) section.
- A new algorithm type named **C1-BNFTPRCF** is introduced in this release. It enables you to prorate the membership premium and non-premium benefit charges for the fully insured group and fully insured individual businesses.

Fully Insured Group Policy

The following changes are made to the Fully Insured Group Policy feature:

- If the advance deposit level for a parent customer is set to **Policy**, the advance deposit details, such as advance deposit applicability, advance deposit amount, advance deposit grace days, and advance deposit account are displayed in the **Main** section of the **Policy** zone. Note that the **Advance Deposit Amount**, **Advance Deposit Grace Days**, and **Advance Deposit Account** fields appear only when the **Advance Deposit Applicability** option is selected.

Framework Upgrade

Oracle Utilities Application Framework Version is upgraded from 4.5.0.0.0 to 4.5.0.1.1 in Oracle Revenue Management and Billing Version 6.0.0.0.0. For more information about the new features and enhancements introduced in Oracle Utilities Application Framework (OUAF) Version 4.5.0.1.1, refer to the [Oracle Utilities Application Framework Version 4.5.0.1.1 Enhancements](#) section. For more information about the deprecations made in Oracle Utilities Application Framework (OUAF) Version 4.5.0.1.1, refer to the [Deprecation Notices for OUAF Version 4.5.0.1.1](#) section.

Redwood User Experience

Oracle Revenue Management and Billing Version 6.0.0.0.0 is certified to use Oracle's Redwood user experience. By default, the value of the **User Interface Style** menu option (i.e. USER_INTERFACE_STYLE) is set to **REDWOOD** during the ORMB installation. The Old-style user experience (i.e. OPE) is no longer supported and should not be used anymore.

Menu Item Search

The **Search** field in the **Application** toolbar allows you to search for a menu item to navigate directly to the corresponding page or BPA script rather than using the menus to navigate to the desired page or script. While searching for a menu item, you need to enter text starting with a slash "/" to indicate that the search is confined to menu items.

Upgrade Impact

Until now, whenever the data of an array list (for example, entity characteristic list, entity division list, entity cost list, etc.) was updated, the system used to set the **actionFlag** attribute of the corresponding data element, all other data elements of the array list (irrespective of whether they are modified or not), and the array list to 'C'. As a result, the system used to update all the data elements of the array list which degraded the system performance. Now, to address the performance issues while updating the data of an array list (for example, entity characteristic list, entity division list, entity cost list, etc.), the system will do the following:

- Set the **actionFlag** attribute of the data element that is modified in the array list to 'C'
- Set the **actionFlag** attribute of all other data elements that are not modified in the array list to 'NULL'
- Set the **actionFlag** attribute of the array list (i.e. at the root level) to 'NULL'

This framework change has a major impact on the base and custom maintenance objects and business objects. Until now, the system considered the **actionFlag** attribute of the array list whenever a data element of an array list was updated. Now, the system will consider the **actionFlag** attribute of the data element that is updated in the array list.

You will have to make the similar changes in the custom maintenance objects and business objects to ensure that the **actionFlag** attribute of the respective data element in the array list is considered whenever the data in an array list is updated.

User Interface (UI) Level Changes

The following table lists changes made to the existing screens in Oracle Revenue Management and Billing:

Screen Name (in 6.0.0.0.0)	Changes
Offset Request Type (Used for Adding, Editing, and Copying)	The following change is made to this screen: <ul style="list-style-type: none"> A new field named Offset Category is added in the Main section.
Offset Request Type	The following change is made to this screen: <ul style="list-style-type: none"> The Offset Category field is added in the Main section of the Offset Request Type zone.
Offset Request (Used for Viewing)	The following changes are made to this screen: <ul style="list-style-type: none"> The Selected Bills and Offset Request Adjustments zones do not appear while viewing the details of an auto maintenance offset request. Instead, a new zone named Auto Maintenance Financial Transactions appears in the Main tab. The Account Information field appears in the Main section of the Offset Request zone while viewing the details of an auto maintenance offset request. A new zone named Adjustment appears while viewing the details of an advance deposit offset request.
Health Product 360° View	The following change is made to this screen: <ul style="list-style-type: none"> The HIOS ID field is newly introduced in the 360° Search zone when you search using the Health Plan Details option from the Search By list.
Health Plan 360° Information	The following change is made to this screen: <ul style="list-style-type: none"> The HIOS ID field is added in the Plan Information zone. It appears only when the HIOS ID is available for the health plan.
Reconciliation Type (Used for Adding and Editing)	The following change is made to this screen: <ul style="list-style-type: none"> A new field named Reconciliation Category is added in the Main section.

Screen Name (in 6.0.0.0.0)	Changes
Reconciliation Type	<p>The following changes are made to this screen:</p> <ul style="list-style-type: none"> • The Reconciliation Category column is added in the Reconciliation Type List zone. • The Reconciliation Category field is added in the Main section of the Reconciliation Type zone.
Reconciliation (Used for Viewing)	<p>The following changes are made to this screen:</p> <ul style="list-style-type: none"> • The Health Plan and Payment Type fields are added in the Pay Instructions zone. These fields appear only when you are viewing the details of a reconciliation object which is created for the fully insured individual business. • The Member Identifier Type and Member Identifier fields are renamed to Subscriber Identifier Type and Subscriber Identifier, respectively, in the Pay Instructions zone. • The Member Name, Member Identifier Type, and Member Identifier columns are renamed to Subscriber Name, Subscriber Identifier Type, and Subscriber Identifier, respectively, in the Pay Instructions zone. • The Payor Identifier Type, Payor Identifier, Health Plan, HIOS ID, and Payment Type columns are added in the Pay Instructions zone. These columns appear only when you are viewing the details of a reconciliation object which is created for the fully insured individual business.
Pay Instruction (Used for Viewing)	<p>The following changes are made to this screen:</p> <ul style="list-style-type: none"> • The Health Plan, HIOS ID, Payment Type, Payor Identifier Type, and Payor Identifier fields are added in the Main section of the Pay Instruction zone when you are viewing the details of a pay instruction which is created for the fully insured individual business. • The Member Name, Member Identifier Type, and Member Identifier fields are renamed to Subscriber Name, Subscriber Identifier Type, and Subscriber Identifier, respectively, in the Main section of the Pay Instruction zone.

Screen Name (in 6.0.0.0.0)	Changes
Discrepancy Report (Used for Viewing)	<p>The following changes are made to this screen:</p> <ul style="list-style-type: none"> • The Health Plan, HIOS ID, Payment Type, Payor Identifier Type, and Payor Identifier columns are added in the Discrepancy Line Items zone. These columns appear when you are viewing the discrepancy line item for the fully insured individual business. • The Member Name, Member Identifier Type, and Member Identifier columns are renamed to Subscriber Name, Subscriber Identifier Type, and Subscriber Identifier, respectively, in the Discrepancy Line Items zone.
Health Plan Search (Window)	<p>The following change is made to this screen:</p> <ul style="list-style-type: none"> • The HIOS ID field is added in the Search Criteria section.
Invoice Request (Used for Searching)	<p>The following change is made to this screen:</p> <ul style="list-style-type: none"> • A new option named Membership is added in the External Reference Type list. Depending on the external reference type that is selected, the system will accordingly allow you to search for a policy or individual membership using the Search functionality in the External Reference ID field.
Invoice Request (Used for Viewing)	<p>The following change is made to this screen:</p> <ul style="list-style-type: none"> • The Account Information column does not appear in the Invoice Request Details zone when you are viewing the details of an invoice request which is created for the fully insured individual business. It appears only when the invoice request is created for the fully insured group business.
Business Rule (Used for Searching)	<p>The following change is made to this screen:</p> <ul style="list-style-type: none"> • The Minor Maximum Number of Dependents column is added to the Search Results section when you search for a fully insured pricing business rule in the Search Business Rule zone.
Fully-Insured Pricing Business Rule (Used for Adding and Editing)	<p>The following change is made to this screen:</p> <ul style="list-style-type: none"> • The Minor Maximum Number of Dependents column is added to the grid.

Screen Name (in 6.0.0.0.0)	Changes
Pricing Rule Type (Used for Adding and Editing)	<p>The following change is made to this screen:</p> <ul style="list-style-type: none"> The Age Recalculation Number of Days field is available in the Age Based Specific Additional Data section when you define or edit an age based pricing rule type.
Pricing Rule Type (Used for Viewing)	<p>The following change is made to this screen:</p> <ul style="list-style-type: none"> The Age Recalculation Number of Days field is added in the Age Based Specific Additional Data section of the Pricing Rule Type zone.
Business Rule Type (Used for Viewing)	<p>The following changes are made to this screen:</p> <ul style="list-style-type: none"> The Main, Parameters, Characteristics, Record Actions, and Record Information sections are added to the Business Rule Type zone.
Parameter (Used for Searching)	<p>The following changes are made to this screen:</p> <ul style="list-style-type: none"> The following options are newly added in the Source Entity list - Customer Class, Delinquency Event Type, and Division.
Parameter (Used for Viewing)	<p>The following changes are made to this screen:</p> <ul style="list-style-type: none"> A new option named Business Rule Eligibility Criteria is added to the Parameter Usage section. It appears only when the source entity is set to either of the following: <ul style="list-style-type: none"> Account Customer Class Delinquency Event Type Division Health Plan Health Product Membership Membership Person Policy Policy Plan

Screen Name (in 6.0.0.0.0)	Changes
Parameter (Used for Adding and Editing)	<p>The following changes are made to this screen:</p> <ul style="list-style-type: none"> • A new option named Business Rule Eligibility Criteria is added to the Parameter Usage section. It appears only when the source entity is set to either of the following: <ul style="list-style-type: none"> ○ Account ○ Customer Class ○ Delinquency Event Type ○ Division ○ Health Plan ○ Health Product ○ Membership ○ Membership Person ○ Policy ○ Policy Plan
Business Rule (Used for Searching)	<p>The following changes are made to this screen:</p> <ul style="list-style-type: none"> • The Delinquency Grace Period Business Rule, Delinquency Event Attributes Business Rule, Delinquency Termination Date Rule Business Rule, and Delinquency Miscellaneous Options Business Rule options are added in the Search By list.
Business Rule (Used for Adding)	<p>The following changes are made to this screen:</p> <ul style="list-style-type: none"> • The Delinquency Grace Period Business Rule, Delinquency Event Attributes Business Rule, Delinquency Termination Date Rule Business Rule, and Delinquency Miscellaneous Options Business Rule options are added in the Business Rule Category list. • The following new screens are introduced in this release: <ul style="list-style-type: none"> ○ Delinquency Event Attribute Business Rule ○ Delinquency Grace Period Business Rule ○ Delinquency Miscellaneous Options Business Rule ○ Delinquency Miscellaneous Options Business Rule ○ Delinquency Termination Date Rule Business Rule

Screen Name (in 6.0.0.0.0)	Changes
	Each screen appears when the respective category is selected from the Business Rule Category list.
Customer 360° Information	<p>The following changes are made to this screen:</p> <ul style="list-style-type: none"> • The Person Collection section is added in the Person Information zone of the Person tab. • The Skip Auto Maintenance field appears in the Account Information zone of the Account tab when it is set to Y for the respective account. • The Delinquency Process Information column is added in the Open Bills zone of the Account tab. • The Rules column is added in the Bill Group Pricing Groups zone of the Pricing Information tab. • The following fields are added in the Main section of the Policy Information zone: <ul style="list-style-type: none"> ○ Advance Deposit Applicability ○ Advance Deposit Amount ○ Advance Deposit Grace Days ○ Advance Deposit Account
Collection Class	<p>The following changes are made to this screen:</p> <ul style="list-style-type: none"> • The Parental Delinquency and Self-Control Delinquency options are added in the Collection Method list.
Address	<p>The following changes are made to this screen:</p> <ul style="list-style-type: none"> • On clicking the Advanced Search link, a new field named Status appears in the Search Address zone. • The Status column is added in the Search Results section.
Address Information	<p>The following change is made to this screen:</p> <ul style="list-style-type: none"> • The Status field is added in the Main section of the Address zone.
Payment Request (Used for Payment Transfer)	<p>The following change is made to this screen:</p> <ul style="list-style-type: none"> • The Payor Account Information field is added in the Main section.
Membership	<p>The following change is made to this screen:</p> <ul style="list-style-type: none"> • The Reprice After Date field is added in the Main section of the Membership zone. It appears only when you are viewing the details of an individual membership.

Screen Name (in 6.0.0.0.0)	Changes
Payment Event Summary	<p>The following change is made to this screen:</p> <ul style="list-style-type: none"> • A new field named External Reference ID appears when you search for a payment event using the Tender Details option from the Search By list.
Price List Assignment (which is accessible from the Deal Information screen)	<p>The following change is made to this screen:</p> <ul style="list-style-type: none"> • The Characteristics section is added in the Assign Price List section.
Benefit Sub Type (Used for Defining, Editing, and Copying)	<p>The following change is made to this screen:</p> <ul style="list-style-type: none"> • The Premium Benefit option is added in the Main section.
Benefit Sub Type	<p>The following change is made to this screen:</p> <ul style="list-style-type: none"> • The Premium Benefit option appears in the Main section of the Benefit Sub Type zone when it is selected while defining the benefit sub type.
Benefit Pricing Rule Type (Used for Defining and Editing)	<p>The following change is made to this screen:</p> <ul style="list-style-type: none"> • The Proration Required column is added in the Price Items section.
Pricing Rule Type	<p>The following change is made to this screen:</p> <ul style="list-style-type: none"> • The Proration Required column appears in the Price Items section of the Pricing Rule Type zone when you are viewing the details of a benefit pricing rule type.
Adjustment	<p>The following changes are made to this screen:</p> <ul style="list-style-type: none"> • Two new fields named Linked Entity Type and Linked Entity ID appears in the Main tab when you are viewing the details of an underpayment or short payment adjustment created for a bill.
Payment	<p>The following changes are made to this screen:</p> <ul style="list-style-type: none"> • The Payment Related Adjustments grid appears in the Main tab when the underpayment or short payment adjustments are directly linked to a payment. • The Adjustment ID column is renamed to Adjustment Information in the Payment Segments and Adjustments grid on the Main tab. • The Adjustment ID column is renamed to Adjustment Information in the Contracts grid on the Pay Segments tab.

Screen Name (in 6.0.0.0.0)	Changes
Policy	<p>The following changes are made to this screen:</p> <ul style="list-style-type: none">• The following fields are added in the Main section of the Policy zone:<ul style="list-style-type: none">○ Advance Deposit Applicability○ Advance Deposit Amount○ Advance Deposit Grace Days○ Advance Deposit Account
Diagnostic Central	<p>The following changes are made to this screen:</p> <ul style="list-style-type: none">• The Trial Bill field is removed from the Search Anomaly zone.• New values appear in the Anomaly Reason list when you search for an anomaly at the bill or bill segment level.

Database Level Changes

This section highlights the documents that you can refer for the following database level changes:

- [New Objects in the ORMB V6.1.0.0.0 Database](#)
- [New Objects in the OUAF V4.5.0.1.1 Database](#)
- [New Tables in the ORMB V6.1.0.0.0 Database](#)
- [New Tables in the OUAF V4.5.0.1.1 Database](#)
- [Existing Tables Modified in ORMB V6.1.0.0.0](#)
- [Existing Tables Modified in OUAF V4.5.0.1.1](#)
- [Algorithms and Algorithm Types Dropped in ORMB V6.1.0.0.0](#)
- [Algorithms and Algorithm Types Dropped in OUAF V4.5.0.1.1](#)
- [Parameters Added or Removed from Algorithm Types in ORMB V6.1.0.0.0](#)
- [Option Types Added or Removed from Feature Configurations in ORMB V6.1.0.0.0](#)
- [Characteristic Types Dropped in ORMB V6.1.0.0.0](#)
- [Batch Controls Dropped in ORMB V6.1.0.0.0](#)
- [Parameters Added or Removed from Batch Controls in ORMB V6.1.0.0.0](#)
- [Parameters Added or Removed from Batch Controls in OUAF V4.5.0.1.1](#)
- [Default User Group Application Services](#)

New Objects in the ORMB V6.1.0.0.0 Database

To view the list of objects (such as tables, columns, algorithm types, business objects, and so on) which are newly introduced in Oracle Revenue Management and Billing Version 6.1.0.0.0, refer to the Appendix A: New Objects in the Oracle Revenue Management and Billing V6.1.0.0.0 Database in the *Oracle Revenue Management and Billing Database Administrator's Guide*.

New Objects in the OUAF V4.5.0.1.1 Database

To view the list of objects (such as tables, columns, algorithm types, business objects, and so on) which are newly introduced in Oracle Utilities Application Framework Version 4.5.0.1.1, refer to the Appendix D: New Objects in the Oracle Utilities Application Framework V4.5.0.1.1 Database in the *Oracle Revenue Management and Billing Database Administrator's Guide*.

New Tables in the ORMB V6.1.0.0.0 Database

To view detail information about the tables that are newly introduced in Oracle Revenue Management and Billing Version 6.1.0.0.0, refer to the Appendix C: New Tables Added in ORMB Version 6.1.0.0.0 in the *Oracle Revenue Management and Billing Upgrade Guide*.

New Tables in the OUAF V4.5.0.1.1 Database

To view detail information about the tables that are newly introduced in Oracle Utilities Application Framework Version 4.5.0.1.1, refer to the Appendix M: New Tables Added in OUAF Version 4.5.0.1.1 in the *Oracle Revenue Management and Billing Upgrade Guide*.

Existing Tables Modified in ORMB V6.1.0.0.0

To view the columns that are newly added, modified, or dropped from the existing tables in Oracle Revenue Management and Billing Version 6.1.0.0.0, refer to the Appendix D: Existing Tables Modified in ORMB Version 6.1.0.0.0 in the *Oracle Revenue Management and Billing Upgrade Guide*.

Existing Tables Modified in OUAF V4.5.0.1.1

To view the columns that are newly added, modified, or dropped from the existing tables in Oracle Utilities Application Framework Version 4.5.0.1.1, refer to the Appendix N: Existing Tables Modified in OUAF Version 4.5.0.1.1 in the *Oracle Revenue Management and Billing Upgrade Guide*.

Algorithms and Algorithm Types Dropped in ORMB V6.1.0.0.0

To view the algorithms and algorithm types which are dropped in Oracle Revenue Management and Billing Version 6.1.0.0.0, refer to the Appendix E: Algorithms and Algorithm Types Dropped in ORMB Version 6.1.0.0.0 in the *Oracle Revenue Management and Billing Upgrade Guide*.

Algorithms and Algorithm Types Dropped in OUAF V4.5.0.1.1

To view the algorithms and algorithm types which are dropped in Oracle Utilities Application Framework Version 4.5.0.1.1, refer to the Appendix O: Algorithms and Algorithm Types Dropped in OUAF Version 4.5.0.1.1 in the *Oracle Revenue Management and Billing Upgrade Guide*.

Parameters Added or Removed from Algorithm Types in ORMB V6.1.0.0.0

To view the parameters which are newly added or dropped from the existing algorithm types in Oracle Revenue Management and Billing Version 6.1.0.0.0, refer to the Appendix F: Parameters Added or Removed from Algorithm Types in ORMB Version 6.1.0.0.0 in the *Oracle Revenue Management and Billing Upgrade Guide*.

Option Types Added or Removed from Feature Configurations in ORMB V6.1.0.0.0

To view the option types which are newly added or dropped from the existing feature configurations in Oracle Revenue Management and Billing Version 6.1.0.0.0, refer to the Appendix G: Option Types Added or Removed from Feature Configurations in ORMB Version 6.1.0.0.0 in the *Oracle Revenue Management and Billing Upgrade Guide*.

Characteristic Types Dropped in ORMB V6.1.0.0.0

To view the characteristic types which are dropped in Oracle Revenue Management and Billing Version 6.1.0.0.0, refer to the Appendix H: Characteristic Types Dropped in ORMB Version 6.1.0.0.0 in the *Oracle Revenue Management and Billing Upgrade Guide*.

Batch Controls Dropped in ORMB V6.1.0.0.0

To view the batch controls which are dropped in Oracle Revenue Management and Billing Version 6.1.0.0.0, refer to the Appendix I: Batch Controls Dropped in ORMB Version 6.1.0.0.0 in the *Oracle Revenue Management and Billing Upgrade Guide*.

Parameters Added or Removed from Batch Controls in ORMB V6.1.0.0.0

To view the parameters which are newly added or dropped from the existing batch controls in Oracle Revenue Management and Billing Version 6.1.0.0.0, refer to the Appendix J: Parameters Added or Removed from Batch Controls in ORMB Version 6.1.0.0.0 in the *Oracle Revenue Management and Billing Upgrade Guide*.

Parameters Added or Removed from Batch Controls in OUAF V4.5.0.1.1

To view the parameters which are newly added or dropped from the existing batch controls in Oracle Utilities Application Framework Version 4.5.0.1.1, refer to the Appendix P: Parameters Added or Removed from Batch Controls in OUAF Version 4.5.0.1.1 in the *Oracle Revenue Management and Billing Upgrade Guide*.

Default User Group Application Services

Oracle Revenue Management and Billing provides the following default user groups - ALL_SERVICES, C1_BSERVICES, HCADMIN, and INADMIN. To view the application services configured for the default user groups, refer to the Appendix B: Application Services Configured for Default User Group in the *Oracle Revenue Management and Billing Database Administrator's Guide*.

Supported Platforms

The following table lists the operating system and application server combinations on which Oracle Revenue Management and Billing Version 6.1.0.0.0 is supported:

Operating System and Web Browser (Client)	Operating System (Server)	Chipset	Application Server	Database Server
Microsoft Windows 10 (64-bit) with Chromium Edge v108+, Mozilla Firefox ESR 102+, or Google Chrome Enterprise Version 108+ Note: Here, x represents the vendor supported version.	AIX 7.2 TL5+ (64-bit) Note: Version numbers suffixed with "+" are the MINIMUM version supported. That version and all future 4 th digit updates will be supported.	POWER 64-bit	Oracle WebLogic 12.2.1.4 (64-bit)	Oracle Database Server 19c
	Oracle Linux 8.x (64-bit)	x86_64	Oracle WebLogic 12.2.1.4 (64-bit) IBM WebSphere 9.0.5.6	Oracle Database Server 19c
	Red Hat Enterprise Linux 8.x (64-bit) Note: Oracle Revenue Management and Billing is tested and certified on Oracle Linux 8.x. Oracle Linux is 100% user space-compatible with Red Hat Enterprise Linux, and therefore Oracle Revenue Management and Billing is supported on Red Hat Enterprise Linux.	x86_64	Oracle WebLogic 12.2.1.4 (64-bit) IBM WebSphere 9.0.5.6	Oracle Database Server 19c

Operating System and Web Browser (Client)	Operating System (Server)	Chipset	Application Server	Database Server
	Microsoft Windows Server 2016+ (64-bit)	x86_64	Oracle WebLogic 12.2.1.4 (64-bit)	Oracle Database Server 19c

Points to Note:

IBM WebSphere certification is pending for Oracle Revenue Management and Billing Version 6.1.0.0.0.

Microsoft Windows Server is not supported for the Production environments. We strongly recommend you install Oracle Revenue Management and Billing (ORMB) on Microsoft Windows platform only for non-production activities, such as User Acceptance Testing (UAT), development setup, and so on.

Technical Recommendations

To improve the overall batch performance on Windows and Linux platforms, we recommend you to make changes in the following files:

File Name	Change From	Change To
hibernate.properties	hibernate.c3p0.timeout = 300	hibernate.c3p0.timeout = 600
threadpoolworker.sh	MEM_ARGS="-Xms512m -Xmx1024m -XX:MaxPermSize=768m"	MEM_ARGS="-Xms512m -Xmx4096m -XX:MaxPermSize=768m"

Supported Upgrades

At present, we support the step-by-step upgrade from Oracle Revenue Management and Billing Version 5.1.0.0.0 to 6.1.0.0.0 for the health insurance domain and the step-by-step upgrade from Oracle Revenue Management and Billing Version 6.0.0.0.0 to 6.1.0.0.0 for the financial services domain. For more information on how to perform step-by-step upgrade, refer to the following documents which are available on OTN or OHC:

- *Oracle Revenue Management and Billing Version 6.1.0.0.0 Upgrade Path Guide*
- *Oracle Revenue Management and Billing Version 6.1.0.0.0 Upgrade Guide*

If you want to directly upgrade the Oracle Revenue Management and Billing database from 2.5.0.1.0 or any later version to 6.1.0.0.0, refer to the *Oracle Revenue Management and Billing Direct Database Upgrade from 2.5.0.1.0 to 6.1.0.0.0*. We recommend you follow the direct upgrade process while upgrading the ORMB database from 2.5.0.1.0 or any later version to 6.1.0.0.0 because it involves a smaller number of steps and saves time.

Unsupported Integrations

We are not supporting Oracle Documaker integration with Oracle Revenue Management and Billing Version 6.1.0.0.0. Alternatively, you can configure the bill and letter reporting feature so that you can generate bills and letters of a customer in the PDF format from ORMB. This reporting feature is built using native capabilities available in ORMB. For more information about the reporting feature, refer to the following documents which are available on OTN or OHC:

- *Oracle Revenue Management and Billing Banking User Guide*
- *Oracle Revenue Management and Billing Bill and Letter Reports Configuration Guide*

Oracle Utilities Application Framework Version 4.5.0.1.1 Enhancements

The following enhancements are made in Oracle Utilities Application Framework (OUAF) Version 4.5.0.1.1:

- [Application Security Enhancements](#)
- [Product Usability Enhancements](#)
- [Data Export Enhancements](#)
- [To Do Management and Processing Enhancements](#)
- [Batch Processing Enhancements](#)
- [Implementation Tool Enhancements](#)
- [Content Migration Assistant \(CMA\) Enhancements](#)
- [Miscellaneous Enhancements](#)

Note: The **Steps to Enable**, **Tips and Considerations**, **Key Resources**, and **Role Information** sections provide guidelines for enabling each feature, wherever applicable.

Application Security Enhancements

This section describes the following new and enhanced application security features introduced in this release:

- [Detailed Description Added to Application Service](#)
- [Screen Information Hidden on Session Timeout](#)

Detailed Description Added to Application Service

Detailed Description was added as an optional attribute for application services. For most product delivered application services, the **Secured Objects** zone, which highlights which metadata objects in the system are linked to the application service, provides enough information to security administrators for them to understand what features are controlled by this record. There are a small number of application services that are referenced from JavaScript or Java, which are not discoverable by the **Secured Objects** zone. For these types of application services, a detailed description helps provide more information about the purpose and use of the record. For these examples and any other use case where the product has determined that additional information is warranted, the detailed description for such application services includes a detailed description.

Screen Information Hidden on Session Timeout

When a user's session times out, the underlying page is changed so that the underlying information is no longer visible. The user sees an alert that the session has timed out and they should click **OK** to log into the application again.

This ensures that possibly sensitive information is no longer visible.

Product Usability Enhancements

This section describes the following new and enhanced product usability features introduced in this release:

- [Algorithm Portal](#)
- [Application Security Query Portal](#)
- [Application Service Query Portal - Search by Release Version](#)
- [Characteristic Type Portal - Support Large Number of Characteristic Values](#)
- [Currency External Reference](#)
- [Currency Portal](#)
- [Dashboard Location Values Adjustment](#)
- [Display Icon Portal](#)
- [Geographic Type Portal](#)
- [Improved Field Portal](#)
- [Inbound Web Service Query Portal - Additional Search Options](#)
- [Language Portal](#)
- [Lookup Portal](#)
- [Menu Portal](#)
- [Online Help Opens in Oracle Help Center for Oracle Utilities Application Framework Based Applications](#)
- [Phone Type Portal](#)
- [Shortcut Key for Navigating Multiple Tabs Adjusted](#)
- [Unified Search Improvements](#)
- [User Group Portal](#)
- [Work List Zone - Improved Item Navigation](#)

Algorithm Portal

The **Algorithm** page has been converted to a portal, leveraging a more flexible and extendable user interface metaphor. The portal includes a **References** tab that list all entities associated with an algorithm. Knowing where an algorithm is used provides you with a better view of your custom rules and will help you target your testing. This portal improves efficiency for building and maintaining algorithms for developers without impacting extensions.

Application Security Query Portal

While application security can be configured at the granular level of an application service and its access modes, it is more intuitive to review access rights to broader application components such as menu lines, dashboard zones, batch processes, and more. A new **Application Security Query** portal supports various options for reviewing this configuration across users, user groups, and various types of secured components. This portal improves efficiency for building and maintaining security without impacting customizations.

Application Service Query Portal - Search by Release Version

The **Application Service Query** portal is enhanced to support a new query option that lists application service by the release they were introduced in. The recording of this information is only available from this release going forward, so it is only applicable to new application service going forward. This makes the uptake step of reviewing new application services easier. This change is transparent to customizations as it improves efficiency for building and maintaining security.

Characteristic Type Portal - Support Large Number of Characteristic Values

The maintenance of the list of characteristic values for a pre-defined characteristic type has been moved to a separate zone on the portal rather than a list built into the main section. The new zone better handles use cases where a large number of characteristic values exist. The zone includes paging and supports filters for more easily finding and maintaining individual entries.

View and maintain characteristic values for a pre-defined characteristic type in a separate zone with filters and paging support.

Currency External Reference

Note: This feature is only an infrastructure change, allowing for edge products or implementations to choose to add support for currencies that are more than three characters. There is no additional functionality provided by the product for this feature.

There is a new external reference column in the Currency table. This allows implementations to define currencies that use a currency code that is more than three characters. This may be required to support crypto currencies with codes of more than three characters. Internalization currencies and the product's current code field are still three characters.

Be aware of the following:

- The currency table still limits the primary key of currency code to three characters. To support a crypto currency with more than three characters, a unique Currency Code should be defined and the External Reference should be used to capture the recognized industry reference.
- References to the currency in product tables continues to use the three character unique currency code.
- Specific interfaces related to payments may need to be enhanced to support this feature. Contact your implementation team to confirm what customization are needed to handle larger currency references.

Currency Portal

The **Currency** page has been converted from the "list" maintenance fixed page style to a portal-driven user interface page. An "all-in-one" metaphor is used, providing a list of the existing currency codes along with the standard ability to broadcast, edit, delete, duplicate, and add a new record.

The **Currency** page has been converted to a portal to provide you with a more flexible and extendable user experience.

Dashboard Location Values Adjustment

The dashboard location values were adjusted to reflect the behavior in a right-to-left language. The values are now 'Before' and 'After', which reflect the location of the dashboard relative to the location of the main page display.

- If the user's language is 'left to right', then the value of 'Before' means the dashboard is on the left and the value of 'After' means the dashboard is on the right.
- If the user's language is 'right to left', then the value of 'Before' means the dashboard is on the right and the value of 'After' means the dashboard is on the left.

Previously, the values were 'Left' and 'Right' with the product behavior based on a 'left to right' language. For a user with a 'right to left' language, the product was reversing the order of the page, including the location of the dashboard. The updated values now reflect the behavior for both types of languages.

Display Icon Portal

Maintain display icon information using a standard portal.

The **Display Icon Reference** page has been converted to a portal to provide you with a more flexible and extendable user experience.

The display icon reference page has been converted from the 'list' maintenance fixed page style to a portal-driven user interface page. It includes a query zone where you can search by the code or the description. The image of the icon is displayed in the results.

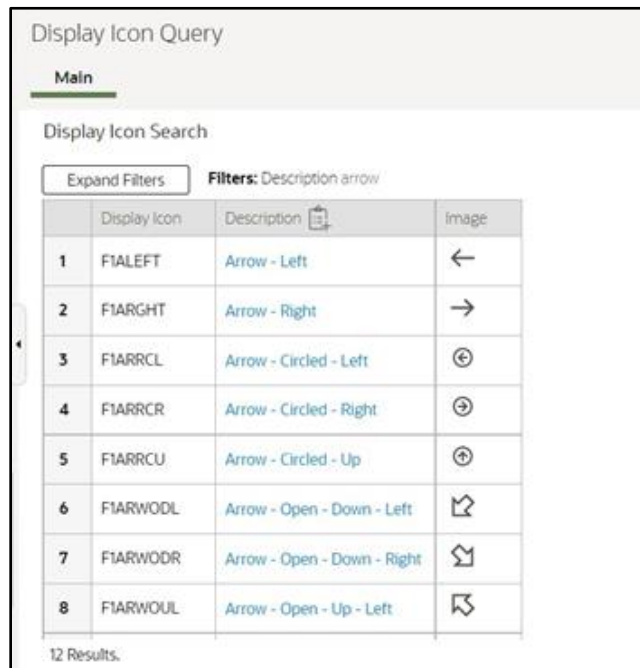


Figure 1: Display Icon Query Screen

Any page that includes a search for a display icon, such as Tree or Insight Type, uses the same query and also benefits from the enhancement to include the icon's image.

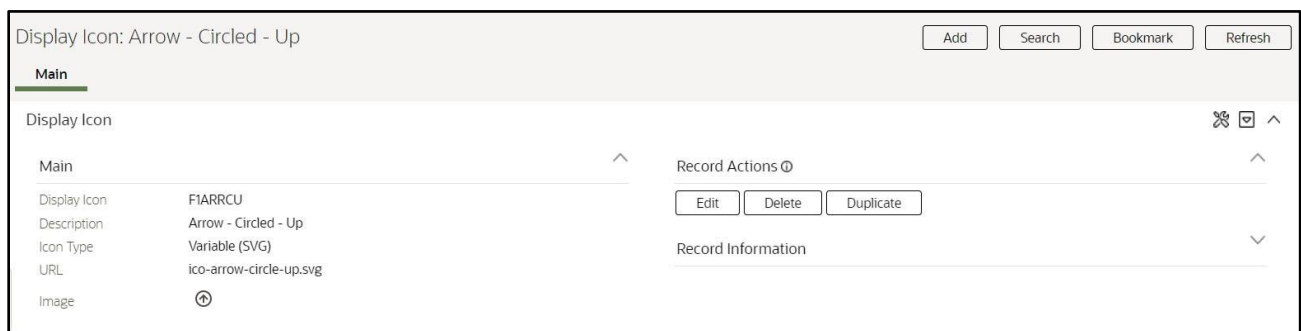


Figure 2: Display Icon View Screen

Geographic Type Portal

The **Geographic Type** page has been converted to a portal to provide you with a more flexible and extendable user experience. An "all-in-one" metaphor is used, providing a list of the existing geographic types along with the standard ability to broadcast, edit, delete, duplicate, and add a new record. This portal improves productivity for adding and maintaining geographical types without impacting extensions.

Improved Field Portal

The **Field** page can now add or update the corresponding lookup field record. The only remaining reason to navigate to the **Lookup** page is for deletion or revision control purposes. Before this release, defining a lookup field involved setting the lookup field itself in the **Lookup** page and then adding it again in the **Field** page. This provides you with a seamless experience to maintain lookup fields. It improves efficiency for developers without impacting customizations.

Inbound Web Service Query Portal - Additional Search Options

The **Inbound Web Service Query** portal is enhanced to support search by information related to REST operations and help text configuration. This provides additional query options for searching inbound web services. This is transparent to customizations as it improves efficiency for building and maintaining inbound web services for developers.

Language Portal

The **Language** page has been converted from a "list" maintenance fixed page style to a portal-driven user interface page. An "all-in-one" metaphor is used, providing a list of the existing language codes along with the standard ability to broadcast, edit, delete, duplicate, and add a new record.

The **Language** page has been converted to a portal to provide you with a more flexible and extendable user experience.

Lookup Portal

The **Lookup** page has been converted to a portal, leveraging a more flexible and extendable user interface metaphor. The portal supports navigating to the corresponding field associated with the lookup. This makes it easier to access the information. This portal improves efficiency for building and maintaining lookups for developers without impacting extensions.

Menu Portal

The **Menu** page has been converted from the fixed page style to a portal-driven user interface page. The **Menu** portal includes an **Application Security** tab for reviewing and configuration application security information for any of the menu lines.

This functionality was previously released on a separate **Menu Application Security** portal that was accessible via a dashboard zone associated with the fixed style menu page. With the conversion of the menu page to a portal, all menu configuration options, including application security setup, are provided in one place.

Online Help Opens in Oracle Help Center for Oracle Utilities Application Framework Based Applications

Opening online help from a Framework-based application now brings you to the relevant documentation page in Oracle Help Center. This removes the need to manage and deploy the help engine used in past releases, and reduces the overall deployment times significantly.

You can use standard browser functions to bookmark pages and page headings, and use your browser's option to open the content in a new tab or open it in a new window. Note that because the help launches as a standard browser window, the default behavior is that it will open in a new tab in the same window. Once the help is launched, you can set in your browser to move the tab location or separate it to its own window as desired going forward. There were also some enhancements to the online help to make navigating the content easier.

Phone Type Portal

The **Phone Type** page has been converted to a portal to provide you with a more flexible and extendable user experience. An "all-in-one" metaphor is used, providing a list of the existing phone types along with the standard ability to broadcast, edit, delete, duplicate, and add a new record. This portal improves productivity when adding or maintaining phone types without impacting extensions.

Shortcut Key for Navigating Multiple Tabs Adjusted

The shortcut key for navigating from one tab of a page to the next tab on the same page is now **Alt+F2**. In the following example, if you are on the **Main** tab of the **User** page and wish to go to the **To Do Roles** tab, you could use **Alt+F2** to get there.

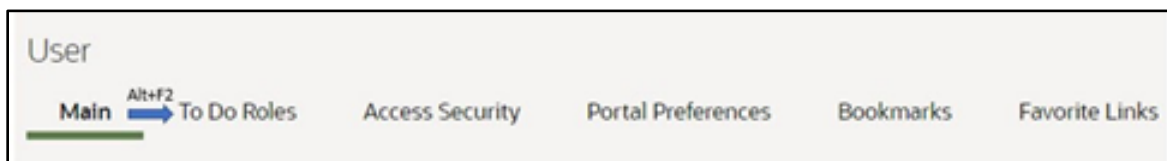


Figure 3: Navigating to the Next Tab

To go to the previous tab, use **Shift+Alt+F2**:



Figure 4: Navigating to the Previous Tab

Previously, the shortcut keys for this functionality were **F2** and **Shift+F2**, respectively. OJet uses **F2** for some of its keyboard navigation and since the product uses OJet widgets for base functionality like Trees and Insights, the shortcut keys were adjusted to be able to take advantage of the built-in OJet features.

Unified Search Improvements

Previously, the Unified Search supported a single search option for a user. Now a user may switch between search options to which they have access. In addition, information about the current search option, which filters and hints it supports, and a recent search history, are all new features easily available to assist the user.

The unified search is enhanced to provide help on each filter and hint, and allow you to toggle between multiple search options you can access.

User Group Portal

The **User Group** page was replaced with a standard query portal that supports additional search filters by application service and user.

The **User Group Maintenance** page was replaced with a standard portal with the ability to use mass actions to:

- Add and remove application services, both at the service and access mode levels.
- Add and remove users.
- Set the expiration data for application services and users.
- Configure security type authorization information.

You can also review application components that are secured by the user group. This is similar to the **Application Security Query** portal which was also introduced in this release.

Work List Zone - Improved Item Navigation

The **Work List** zone is populated whenever you click the **Work List** icon in the header of a data explorer results list.

The following enhancements were made to the **Work List** zone:

- The entry that you are currently working on is marked with a star. Previously, a check was used for both the current entry and entries already visited. Now the check is only for the previously visited entries.

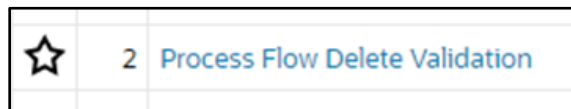


Figure 5: Entry Marked with a Star in the Work List Zone

- Previous** and **Next** buttons allow you to move through the list without having to find the next entry to click.



Figure 6: Previous and Next Buttons

- You may actively mark a row to ignore when using the **Previous** and **Next** buttons. You can do this by clicking the column adjacent to that entry. Clicking this icon removes the indication and allows the entry to be included in the next/previous processing again.



Figure 7: Removing the Indication of an Entry

- If you have a long list, the zone keeps the current entry position in view even after refreshing the dashboard. Previously, any refresh of the zone would reposition the list to the top requiring you to scroll to find your current entry.

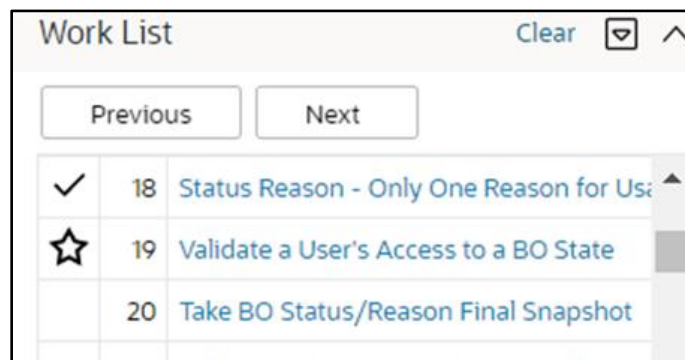


Figure 8: Work List Zone

Data Export Enhancements

This section describes the following new and enhanced data export features introduced in this release:

- [Data Redaction Rules](#)
- [General Data Export - Restrict Initial Export by Time](#)
- [Generalized Export Dashboard Portal Enhancements](#)
- [Generalized Data Export Support for Entity Log Records](#)
- [Generalized Data Export Support for Lost Ongoing Export Files](#)

Data Redaction Rules

Personal Identifiable Information (PII) that is not encrypted may require special handling when it is exported out of the application and written to files. Depending on how the export file is used, this type of information may need to be anonymize before being written to the file. Now, Generalized Data Export and Content Migration Assistant (CMA) batch processes support the anonymization of PII data based on customer- defined redaction rules.

A redaction rule defines the type of information that should be redacted, where it resides, and what function to apply to the value to anonymize it. For example, a phone number field may be redacted by applying a function that replaces all digits with 0. A library of common redaction functions is provided as an extendable lookup, and you may extend the list as needed. Sample redaction rules are provided as accelerator data, and you can update the rules as needed.

As a way of tracking and identifying fields that may contain sensitive information, a new searchable PII Class option now appears in the Field record definition.

The export tools were enhanced as follows:

- **Content Migration Assistant -**
 - By default, data redaction rules are applied to anonymize sensitive information before it is written to the export file.
 - Depending on application security, you may indicate on a specific migration data set export request that data should be exported as it is (non-redacted). This option is only available to if you are provisioned with the "Do Not Apply Redaction Rules (NRDT)" access mode associated with the migration data set export's application service (F1MIGRDEXP).
- **Generalized Data Export Batch Processes -**
 - By default, data redaction rules are applied to anonymize sensitive information before it is written to the export file.
 - A new batch parameter "Do Not Apply Redaction Rules" allows data to be exported as it is (non-redacted). The option requires that the user submitting the batch process is provisioned with the "Do Not Apply Redaction Rules (NRDT)" access mode associated with the data export application service (F1-DATAEXPORT).

- Note that Generalized Data Export data export is a single stream of files that may be sent to multiple targets. You should review these integrations to be sure they do not have conflicting redaction requirements.

Once redaction rules are defined in an environment, these processes will automatically anonymize data by default.

A new **Obfuscation Configuration** portal provides a consolidated view of all masking, encryption, and redaction rules.

General Data Export - Restrict Initial Export by Time

By default, the initial export batch process exports the entire data for an entity. In some situations, typically around high volume historical data, there is a need to restrict the export to more recent data (for example, the last few months of data).

The initial export batch process now supports an optional batch parameter named "Restrict By Date" that allows utilities to constrain the history they export. The parameter references the date field to use and the requested time frame in terms of number of days prior to business date. Refer to the Generalized Initial Export Template (F1-GEIXP) batch control for more information.

Note that excluding records from export should be carefully considered as this may result in data discrepancies when the excluded data is referenced on other entities. Refer to online documentation for additional considerations when using this option.

The option is only applicable to specific entities that have a qualifying reference date and the time frame for export may vary for each customer. Therefore, the new parameter is not added to any base product Initial Export batch control aside of the template batch control for documentation purposes. Customers can clone the base product batch control for a specific entity and add the parameter to their custom version.

Generalized Export Dashboard Portal Enhancements

When the list of entities to add or edit export settings for is too large, the transaction may time out. This situation often required the user to repeat the update in smaller batches. Now, by default, the application attempts to make the update in smaller batches. This minimizes timeout situations and makes them less problematic.

In addition, in some rare situations, the process to enable an entity for ongoing export may leave the entity not fully configured. Entities that are not fully configured are now explicitly highlighted as not configured, and they are listed at the top of the **Export Entities** zone. The user can then repeat the enablement action for these entities to resolve the problem.

Generalized Data Export Support for Entity Log Records

By default, the Generalized Data Export batch process does not export maintenance object log tables due to performance and data volume considerations. Also, log records are omitted from export because they do not usually contain useful business information. However, there are entities for which log records have analytical value. As of this release, a new **Export Log Tables** maintenance object option allows an entity to be configured to explicitly include its log records when exported.

Generalized Data Export Support for Lost Ongoing Export Files

The Generalized Ongoing Export (F1-GEEEXO) batch process now keeps a backup of all entity keys exported by a specific batch run number in a designated backup table. A new Restore Generalized Export Keys (F1-GERST) batch process assists in recovering from such lost file situation. The new process restores the keys of all entities that were exported during a specific batch run, from the backup table back to the ongoing changes queue, causing them to be exported again on the next ongoing export batch run.

Note that this method does not restore the exact content of the original file that was lost. That content cannot be reproduced. Instead, this process ensures that the entities that were included in the lost file would be extracted again.

Previously, the Generalized Ongoing Export (F1-GEEEXO) batch process did not keep a record of which entities were exported on a specific batch run. In cases where an export file is lost or damaged, there was no way to identify which entities should be exported again.

To Do Management and Processing Enhancements

This section describes the following new and enhanced To Do features introduced in this release:

- [To Do Management by Assigned User](#)
- [To Do Dashboard Details Tab](#)

To Do Management by Assigned User

Previously, the **To Do Management** portal restricted queries to a specific To Do Type. Now either a To Do Type or an Assigned User are required. As a result, there is now support for management of entries assigned to a user across multiple to do types.

To Do Dashboard Details Tab

The **To Do Dashboard** portal is enhanced to provide a detailed list of the entries selected by the charts on a new **Details** tab.

The new tab supports further filtering and actions a user can take to manage the selected entries. This provides the same functionality supported by the **To Do Management** portal.

Note: With this enhancement to the **To Do Dashboard**, we strongly recommend to discontinue the use of the **To Do Summary** and **Supervisor To Do Summary** pages. The To Do Dashboard provides the functionality found in the two legacy 'summary' pages, and it also provides additional functionality not found in those pages.

Batch Processing Enhancements

This section describes the following new and enhanced batch processing features introduced in this release:

- [Batch Logs for Issues Found Before Thread Execution](#)
- [Batch Submission Parameters Defaulted from Batch Control](#)
- [Manifest File Includes Previous Manifest File Information](#)
- [Plug-in Driven Batch Variables](#)
- [Plugin-driven Extract - Support for Writing to Multiple Files and Flexibility in File Names](#)
- [Submit Batch Job REST Linux Script](#)
- [Support for Encrypted Files and Digital Signatures](#)
- [Support Override of Base Algorithm on Batch Control](#)

Batch Logs for Issues Found Before Thread Execution

The product captures a standard output file for each thread. If there are errors found in any thread, an error file is also produced which is accessible from the **Batch Run Tree** page. You can now capture log files for steps performed prior to the thread execution, for example in the "get job work" step.

The batch framework writes some basic logging information for the standard output file. The format of the file name produced is *batch control + run number + re-run number + date-time.stdout*. You will see an 'stdout' file for the batch run as well as one for each thread. Individual batch jobs may now also write log statements to the new stdout file at the batch run level.

If a batch job encounters an error in the steps before the thread execution, for example in the "Select Records" step, an error file is also produced. The format is *batch control + run number + re-run number + date-time.stderr*.

These files are also accessible from the **Batch Run Tree** page.

Batch Submission Parameters Defaulted from Batch Control

When submitting a batch job via any method, the system compares the provided parameters to those defined on the batch control.

- Any parameters that are missing from the request are populated from the batch control. Note that if the parameter is required and a default value is defined on the batch control, that value is used. If the parameter is optional, any default value found on the batch control is ignored. For optional parameters, the default value is only for online defaulting (where the user is able to choose to remove the default).

- In addition to the value, parameters define both a parameter name and sequence. If the parameters in the request only provide parameter name or sequence, the missing field is filled in based on the related batch control's definition. Note that if the request includes both the parameter name and the sequence, the system does not validate that the sequence is correct for the parameter name. The parameter name is the important field for the batch job functionality. The sequence is needed to ensure that the row is unique.

This feature improves the usability of the REST API used to submit a batch job (the 'submit' operation for the F1-SubmitJob inbound web service). Previously, the caller of this API was required to determine the batch parameters from the batch control ahead of time. With this enhancement, callers can submit the request for the batch submission and only supply the parameter name/parameter value for optional parameters where a value is desired or for required parameters when the value differs from the default value on the batch control. Any optional parameters where no value is needed or required parameters where the default value from the batch control is acceptable do not need to be provided.

Manifest File Includes Previous Manifest File Information

When sending the latest data export file using the Generalized Data Extract functionality, the manifest file produced with an extracted file now includes the previous successful batch run number and the previous manifest file name. The receiving systems can use this information to compare with the most recent file received to determine if any expected file is missing.

This allows for quick detection of issues when information is regularly extracted to an external system.

Plug-in Driven Batch Variables

The Plug-in Driven Upload process (template batch control is F1-PDUPL) now supports the following variables in the Processed File Extension parameter in addition to text:

- {RDT} or {runDateTime}
- {RD} or {runDate}
- {RT} or {runTime}
- {BC} or {batchCode}
- {BN} or {batchNumber}
- {RN} or {rerunNumber}
- {TN} or {threadNumber}
- {USR} or {userId}

The parameter must still end with text to streamline the existing system check, which ensures that the name of the file to upload does not include the text included in the Processed File Extension parameter. The User ID used for the parameter is the submission user instead of the "batch user."

Also, the Plug-in Driven Extract process (template batch control is F1-PDBEX) now supports the additional variables {USR} or {userId} as part of the file name. When these variables are specified, the submission user's ID is used.

Plugin-driven Extract - Support for Writing to Multiple Files and Flexibility in File Names

The plug-in driven extract process has been enhanced to allow for the Process Records algorithm to return one or more file names to write the data to. This allows for use cases where the data should be segregated by a business value. It also supports indicating a different file name for each schema returned by the algorithm, allowing for one unit of work to contribute to more than one file. Several use cases informed this enhancement:

- The data being extracted is related to different service providers and a file should be produced for each service provider. In this case, service provider is the unit of work and each unit of work could produce a separate file.
- The implementation has several CIS divisions and when extracting data, for example an asset, the data should be segregated such that each division gets its own file.
- An extract of Person data should result in a file with all the individual person information, a file with all the business person information, and a consolidated file of contact information for all person types. In this case, one call to the Process Records algorithm will result in data written to two different files (individual or business person information and contact information).

If a batch job is multi-threaded, each thread continues to produce its own set of files. For the service provider use case above, a multi-threaded process should still result in a file per service provider. However, for the Division and Person examples, each thread may include files for the same division or person type. The option to concatenate files is still supported and will concatenate files where the file name is the same except for the thread number. The existing limitations for the use of file concatenation still apply. This is documented in the batch control parameter description in the metadata.

Submit Batch Job REST Linux Script

The submitbatchREST.sh Linux script enables implementations to submit batch jobs when using external batch schedulers. This script uses the Submit Operation of the F1-SubmitJob REST service to create an entry in the Batch Job Submission table, which the batch daemon polls to pick up and run the batch job. The script also uses the Get Operation of the F1-SubmitJob REST service to periodically poll the system to retrieve the "Get Batch Job" or "Batch Run" details to provide status updates until the completion of the batch job.

Support for Encrypted Files and Digital Signatures

Exchanging and signing encrypted files requires the sender and recipient to share keys. The system needs to record the public keys provided by external parties and generate public/private key pairs so the public keys may be exchanged when encrypting or signing files.

A new public encryption key ring business object (F1-ExtKeyRing) allows the public keys provided by the recipients of encrypted files to be stored. The key ring also captures the external system ID and external reference of the system receiving the encrypted files. A new encryption key pair business object (F1-InternalEncryptionKey) allows private/ public key pairs to be generated and stored using standard PGP format. The new internal key pair is classified as 'File Encryption / Signing'. Key rings previously classified as 'Signature' are now referred to as 'RSA Signature'. A new flag has been added to key ring to categorize key rings as external, internal, or OAuth.

The lifecycle of a key ring key has been changed to provide more flexibility when rotating keys. Internal generated keys are created in a status of pending. They may be manually activated when the public key has been shared with the applicable third parties. When a external public key is added, the new key is automatically activated and the previously active key is expired. Expired keys may be manually inactivated. Note that inactive keys can no longer be activated again.

Batch processing has been enhanced to implement file encryption and decryption using PGP standards. Digital signatures are also supported, using the standard 'Sign' option. New batch parameters have been introduced to specify the external and internal keys to be used for encryption and/or signing and the file adapter automatically encrypts, decrypts and signs the file data based on the existence of these parameters.

Support Override of Base Algorithm on Batch Control

The following batch control plug-in spots are single algorithm plug-in spots:

- Select Records
- Process Record
- File Upload

If the product provides batch controls with algorithms for any of the above plug-in spots plugged in, you can now override the base algorithm by plugging in your own custom algorithm using a higher sequence. The system uses the algorithm with the highest sequence.

Previously, if a base product batch control was provided with algorithms for any of the above plug-in spots, you were not able to override the batch control algorithm. You could only clone the entire batch control if you wanted to override any of the provided algorithms.

Implementation Tool Enhancements

This section describes the following new and enhanced implementation tools introduced in this release:

- [Action Provided to MO Audit Plug-In Spot](#)
- [Attachment Malware Scan Plug-in](#)
- [Base Product Index Data Visible in Table Portal](#)
- [Business Object Status Reasons REST API](#)
- [Characteristic Mapping Language Genericized](#)
- [Debugging Tools - Consolidated into a Single Button](#)
- [Dropdown Lists REST API](#)
- [Expand Tree REST API](#)
- [Extensions Dashboard Portal](#)
- [HTML Editor Syntax Highlighted](#)
- [HTML Row Header Reference Update](#)
- [Insights REST API](#)
- [Javadocs Viewer](#)
- [Menu Item Configuration for Add Action](#)
- [New Base Display Icon Images](#)
- [Parameter Update for New Language Batch Program](#)

Action Provided to MO Audit Plug-In Spot

The action of a record is now provided to the MO Audit plug-in spot. For business use cases where the algorithm should perform different logic based on whether the record is added, changed, or deleted, this information simplifies algorithm code. Previously, algorithms with this requirement had to determine whether the record was new, changed, or deleted themselves.

MO Audit algorithms now receive the action for the impacted object simplifying logic that may differ based on the action.

Attachment Malware Scan Plug-in

The system supports a malware scan algorithm that is called when adding an attachment. This uses the "Attachment Malware Scan" System Event for on the **Algorithms** tab of the **Installation Options - Framework** portal. If an algorithm is plugged into this spot, it is called every time an attachment is uploaded, regardless of the method. In addition, the product has supplied a base algorithm that calls ClamAV. The ClamAV is not provided. Implementations need to separately install that software. If your implementation prefers to use a different malware scanning software, you can implement your own algorithm and plug it into the Installation Options.

Scanning attachments for malware before they are uploaded allows you to prevent harmful information from getting loaded to the system.

Base Product Index Data Visible in Table Portal

The **Index** tab on the **Table** portal now shows the base product-owned indexes. This allows for developers and implementers to view the information about the product delivered indexes.

Previously, the index information was only visible in the database. Also, in a previous release, metadata tables were added for defining indexes associated with a Table, however the new index metadata tables were not populated with the based product delivered indexes.

Business Object Status Reasons REST API

A new Status Reasons REST API (F1-StatusReasons) allows an external user interface application to retrieve valid reasons to be presented to a user when they perform an action that transitions an entity to a new status.

User interface applications like Visual Builder can use a new API for presenting valid reasons for transitioning an entity to a new status.

Characteristic Mapping Language Genericized

The characteristic mapping maintenance object (MO) is enhanced to allow for future support of mapping identifier values as well. As part of this effort, the maintenance object and the UI portal and many of the related artifacts are renamed to be more generic: Analytics Attribute Mapping.

Characteristic Mapping continues to be a business object for this MO and no changes exist for that functionality.

The newly rebranded Analytics Attribute Mapping table now includes new columns to support mapping of identifier values. There is a new business object visible in this release: F1-IDMapping. This business object captures the data needed to configure mapping of a given identifier type and value defined in a child table of an MO to a user defined field for a target dimension for the parent table of that MO. Note however that the changes to Oracle Utilities Analytics Visualization (OUAV) to support the identifier mapping are not supplied in this release. You may choose to define the metadata for the identifier mapping but they will not be used by OUAV at this time.

Debugging Tools - Consolidated into a Single Button

The various debugging tools that are enabled when the URL includes the `debug=true` parameter have been moved to a slide out panel enabled by a new 'bug' icon. The options visible to you when the debug parameter is turned on depend on whether your implementation is cloud or on-premise and your individual security rights.

In addition, the product has provided a new menu entry in the Help menu: Enable | Disable Debug. If you have access to the security application service for this menu entry, you can turn on or off debug without changing the URL parameter.

Dropdown Lists REST API

A "Dropdown Lists" (F1-DropdownLists) REST API retrieves data needed for building one or more drop-downs. Information for a drop-down includes a list of valid codes and their corresponding descriptions.

The web service supports the following types of drop-down data sources:

- Lookup field
- Extendable lookup
- All records in an admin maintenance object (only single prime key field entities are supported)
- Business service/service script computed list of values

User interface applications like Visual Builder (VB) may use a new API for retrieving data displayed as drop-downs.

Expand Tree REST API

The product provides a comprehensive infrastructure that supports user interface rendering of hierarchical information for a specific context based on configurable tree definitions. For example, it can present a tree view of entities related to an account. Now a new "Expand Tree" REST API (F1-ExpandTree) allows an external user interface application to retrieve hierarchical information for a specific context based on the same tree configuration. The information provided by the web service focuses on presentation content and includes minimal visualization recommendations, allowing the external application to render each tree node as per its own configuration and rules. Since they are application specific, the information also does not include action-related settings.

Extensions Dashboard Portal

The new **Extensions Dashboard** portal provides a high-level summary of all utility- owned entities and extensions made to base product entities. Information is summarized by maintenance object and allows the user to view the specific entities for a specific maintenance object and navigate to the respective page or portal for more information. With this information, utilities can better assess the necessary testing when planning for product upgrades and investigate the removal of extensions when upgrading to reduce costs.

HTML Editor Syntax Highlighted

The HTML editor on the UI Map/Schema tab has been updated to use a syntax highlighter to make it easier to review and update HTML.

A screenshot of a web browser window titled "HTML Editor". The window displays HTML code with syntax highlighting. The code includes a DOCTYPE declaration, an HTML header with a title, and a body with a class attribute and an onload event. It also features an Oracle-specific include tag, a CSS link, and a JavaScript function definition for createChart().

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
2 <html xmlns="http://www.w3.org/TR/xhtml1/strict">
3
4 <head>
5   <title></title>
6 </head>
7
8 <body class="oraZoneMap" onload="createChart()">
9
10 <oraInclude map="f1-0JETLIBS" />
11 <link href="cisDisabled.css" type="text/css" rel="stylesheet" />
12
13 <script type="text/javascript">
14   // 
15   function createChart() {
16     var refreshFunc = null;</pre></div><div data-bbox="386 372 609 389" data-label="Caption"><p>Figure 9: HTML Editor Screen</p></div><div data-bbox="112 401 501 418" data-label="Text"><p>This makes it easier to review and update the HTML.</p></div><div data-bbox="112 435 472 454" data-label="Section-Header"><h2>HTML Row Header Reference Update</h2></div><div data-bbox="112 468 889 521" data-label="Text"><p>The HTML5 standard requires the use of the &lt;TH&gt; element within a table row when it is being used as the row header (&lt;th scope="row"&gt;&lt;/th&gt;). Previously, these elements were allowed to use a &lt;TD&gt; element for the row header.</p></div><div data-bbox="112 532 889 587" data-label="Text"><p>For tables that are generated by Oracle Utilities Application Framework (Data Explorers, standard UI Map grids, and more), the HTML is automatically updated to match Oracle standards. If you are creating a table manually, a change is not required but it is recommend you update your HTML for better accessibility.</p></div><div data-bbox="112 602 280 623" data-label="Section-Header"><h2>Insights REST API</h2></div><div data-bbox="112 636 889 690" data-label="Text"><p>The product already provides a comprehensive infrastructure that supports user interface rendering of various types of insights for specific contexts. For examples, billing insights about an account, person, and more.</p></div><div data-bbox="112 700 889 793" data-label="Text"><p>A new "Insights" REST API (F1-Insights) allows an external user interface application to retrieve insight information for a specific context based on the same insight configuration. The information provided by the web service focuses on insight content and includes minimal visualization recommendation, allowing the external application to render the insights as per its own configuration and rules. For the same reason, the information does not include action-related settings because they are application specific.</p></div><div data-bbox="112 928 145 944" data-label="Page-Footer">236</div><div data-bbox="529 928 888 945" data-label="Page-Footer">Copyright © 2009, 2024, Oracle and/or its affiliates.</div>
```

Javadocs Viewer

Javadocs viewer is now available via the **Admin → Implementation Tools** menu. This launches the Javadocs landing page, which allows you to view information about base delivered Java classes and packages. In addition, when viewing an algorithm type, batch control, or foreign key reference, a 'source viewer' icon is visible when a Java program is displayed. If you click the icon, it launches Javadocs for that particular program.

In the **Script** page, the **Script Tips** zone in the dashboard includes a link to View Groovy Javadocs. This allows script writers to view information about methods and classes available for developing Groovy scripts.




Menu Item Configuration for Add Action

Menu items include optional configuration for application service and access mode. For menu items associated with searching, security is controlled by the associated portal or transaction's application service. Therefore, defining an application service on these menu item is only useful to override the base application service, which is not common. However, for many menu items related to the add action, if a user did not have access to that action, they may still see the option on the menu and would only get a security error later after attempting to add. Utilities could override this by configuring an application service for the "add" Menu Item.

All product menu items handling an 'add' or any other action are configured with an application service or access mode by default.

New Base Display Icon Images

The following additional SVG icons are provided for use in contextual insights, trees, and other user interface features that support SVG icons.

Icon	ID	Description
	F1STAR	Star
	F1STARTS	Star-Solid
	F1VWHD	Hidden from View

Additional icons allow for an enhanced user experience for displayed information.

Parameter Update for New Language Batch Program

The language input parameters in the F1-LANG (New Language) batch program were adjusted to avoid confusion when submitting the batch for deletion of a language.

The two parameters were previously called Source Language and Target Language. Their descriptions have changed to Copy From Language and Action Language. The parameter names were not changed.

* 20	sourceLanguage	Copy From Language	Enter the language code for the source data. Only required for the add action.
* 30	targetLanguage	Action Language	Enter the language code for which the data is being added or deleted.

Figure 10: Parameters for Adding New Language

The behavior for adding a new language has also not changed. You should continue to put the Language to copy from first, and then the language you are adding as the second parameter. For example, if you are creating German language rows, using English as the Copy From Language, you would populate:

- **Action:** ADD
- **Copy From Language (formerly Source Language):** ENG
- **Action Language (formerly Target Language):** GER

However, the behavior for deleting changes has been updated. Previously, you populated the Source Language for the delete. Now you populate the Action Language. For example:

- **Action:** DEL
- **Copy From Language (formerly Source Language):**
- **Action Language (formerly Target Language):** GER

In addition, the batch program has been updated to prevent you from deleting rows for the ENG language that would result in the product system metadata being deleted.

Content Migration Assistant (CMA) Enhancements

This section describes the following new and enhanced CMA features introduced in this release:

- [CMA Import Performance Improvements](#)
- [CMA Support for Separate Configuration and Business Data Migrations](#)
- [Export Content Migration Assistant Data to an Older Version](#)
- [Migration Plan Pre-Compare Algorithm Execution Update](#)

CMA Import Performance Improvements

By default, the Content Migration Assistant (CMA) import process creates a migration object for each imported entity. This allows for a granular reporting and error handling at the entity level. When importing a high volume data set of business entities, this granular management has a performance toll.

The CMA import process now supports a Bulk Import option by which a group of entities, of the same maintenance object, is managed as a single migration object. Using this option reduces the migration object management effort throughout the process and results in better performance. Each entity is still individually validated in this mode, as in regular import processing, but if one entity is invalid the entire migration object is not applied (impacting all entities within it). Bulk import mode is useful when importing a large set of data from a validated data source when almost no errors are anticipated.

The indication of whether to use bulk import or not is specified on the migration data set import request.

Another new option, also specified on the migration data set import request, allows you to indicate that all imported entities are assumed to be new additions to the current environment. By indicating that this import is Insert Only, the import process avoids unnecessary steps to determine whether the entity should be added or updated.

These options are only supported for master and transaction maintenance objects. They are not applicable to configuration migrations.

The import step of the CMA process has been enhanced to execute insert statements more efficiently by leveraging the database's bind variables functionality. This is transparent to existing in-progress migrations.

CMA Support for Separate Configuration and Business Data Migrations

By default, the same import CMA batch processes manage configuration and business data migrations. Typically, business data migrations involve high volume of records compared to much lighter configuration data sets. Processing them together by the same batch process slows down the performance of configuration migrations, preventing them from completing faster and more frequently. The issue mainly affects test-like environments where mixed data class migrations is more common: configuration data is imported from a lower environment and large test data is imported from a higher environment. Now you can adjust your CMA configuration in such environments to benefit from separating the import processes for configuration and business data.

The need to separate import processes mainly applies to migration objects because of their volume. Migration data set and transaction records are of low volume and as such are still managed by the same batch processes.

New batch processes are provided for importing migration objects containing business data:

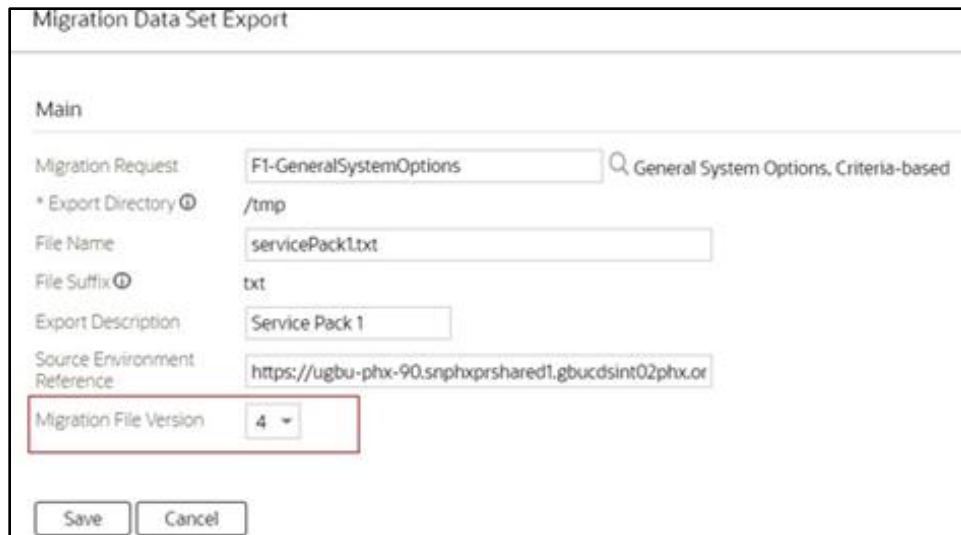
- F1-MGOPB - Migration Object Monitor (Business)
- F1-MGOAB - Migration Object Monitor (Business) - Apply

By default, the existing migration object batch processes continue to handle both classes of data. To use the new batch processes for business data, you need to associate the new business data migration object BO provided as part of this enhancement with the new batch processes. See the steps below for details.

Export Content Migration Assistant Data to an Older Version

When performing a Content Migration Assistant (CMA) export, the format of the file is specific and the CMA import step knows how to read the file. There are times when the product makes an enhancement to CMA that requires a change in the structure of the export file at which point the 'version' of the CMA file is incremented. When this happens, the import step is able to detect if you are importing an older version of the CMA file and proceeds accordingly.

The product now provides support for exporting a CMA file using an older version format. For example, the current CMA version is 5. When creating a Data Set Export, you can indicate that you want to export using version 4. This is useful if you have upgraded one of your environments to a version of the product that uses CMA version 5, and you need to export data to an environment that has not yet been upgraded and is therefore still on version 4.



The screenshot shows a dialog box titled "Migration Data Set Export" with a "Main" tab. The fields are as follows:

Field	Value
Migration Request	F1-GeneralSystemOptions
* Export Directory	/tmp
File Name	servicePack1.txt
File Suffix	txt
Export Description	Service Pack 1
Source Environment Reference	https://ugbu-phx-90.snphxprshared1.gbucdsint02phx.or
Migration File Version	4

Buttons for "Save" and "Cancel" are located at the bottom of the dialog.

Figure 11: Setting the Migration File Version for Data Export

This is analogous to choosing to save a Microsoft Word document using an older version of Word (for example, with the .doc file type) during the period where not all software could consume the newer (.docx) version.

Migration Plan Pre-Compare Algorithm Execution Update

When the migration data import process runs the pre-compare algorithms for a given migration plan, the logic now includes the algorithms configured on the migration plan in the target environment as part of the execution. Previously, this step only included the pre-compare algorithms carried over from the source environment. This ensures that any additional data updates applicable at the target are included.

Miscellaneous Enhancements

This section describes the following new and enhanced miscellaneous features introduced in this release:

- [Application Service's Application Security Zones](#)
- [Cloud Object Storage Support for Platform as a Service \(PaaS\)](#)
- [Digital Self Service Masquerading Using Key Ring](#)
- [HTML Sanitization Improvements](#)
- [Improved Characteristic Type Referential Integrity](#)
- [Key Ring Support for OAuth Client and Secret](#)
- [New Class Element on Business Flag Standard Name Business Object](#)
- [Support for Capturing and Exporting Batch Run Analytics Data](#)

Application Service's Application Security Zones

The zones on the **Application Service - Application Security** tab have the following improvements:

- The **Deny Access** button in the **User Groups Linked** zone has been converted to a mass action button, allowing updates to multiple groups with one action. In addition, the search filters have been enhanced to provide a search within search to find a specific user or user group, filter by user group description, and exclude expired links. The Expiration Date column has been enhanced to highlight expired links in bold with red text and an asterisk. The Access Modes column now shows only the application service access codes that are not configured for the user group.

The screenshot shows the 'User Groups Linked' interface. At the top, there is a 'Deny Access' button. Below it is a table with the following columns: User Group, Description, Expiration Date, Missing Access Modes, and Security Types with Authorization. The table contains four rows of data:

User Group	Description	Expiration Date	Missing Access Modes	Security Types with Authorization
<input type="checkbox"/> ALL_SERVICES	System User Group	01-01-2100		
<input type="checkbox"/> CMA-WAM-BASE	Base User Group ALL non-WAM_BASIC Users would need	01-01-2100	Add,Change,Complete,Execute,Forward,Send Back	
<input type="checkbox"/> CMA-WAM-READ-ONLY	Read Only Access to all non-Admin WAM Portals	01-01-2100	Add,Change,Complete,Execute,Forward,Send Back	
<input type="checkbox"/> IND-DACS-SERVICES	New Digital Asset Cloud Service Security	01-01-2100	Add,Change,Complete,Execute,Forward,Send Back	

Below the table, there are search filters: User, User Group, User Group Description, and Exclude Expired User Groups. A Search button is located at the bottom left.

Figure 12: User Groups Linked Zone

- The **Grant Access** button on the **User Groups Not Linked** zone has been converted to a mass action, allowing updates to multiple groups with one action. A pop-up window provides the ability to set the expiration date and remove unwanted access modes. In addition, the search filters have been enhanced to provide a search within search to find a specific user or user group and to filter by user group description.

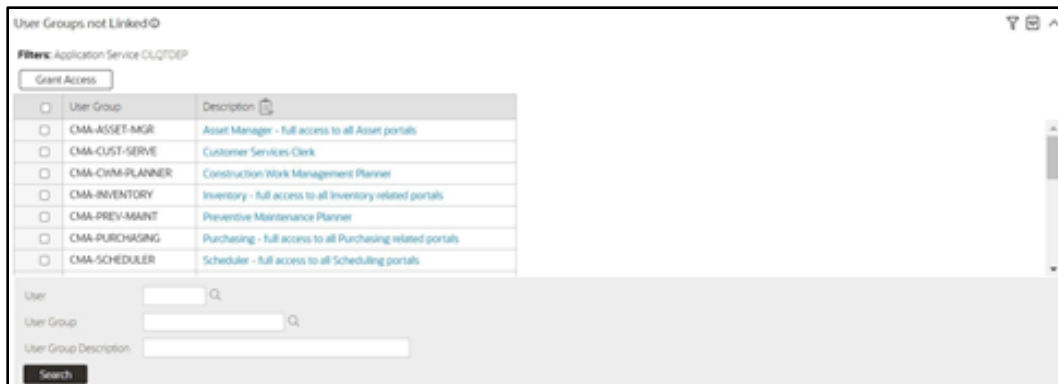


Figure 13: User Groups Not Linked Zone

Cloud Object Storage Support for Platform as a Service (PaaS)

Implementations using Platform as a Service (PaaS) can configure the file adapter of the Oracle Cloud Infrastructure Object Storage service to store interface files and reports.

Digital Self Service Masquerading Using Key Ring

The product supports the ability for a utility to authorize an external person to view the customer's account information in a customer-facing portal (such as Digital Self Service). The system uses a private/public key to enforce standard security measures. Previously, the Redirection Targets master configuration captured a reference to the alias for the key pair stored in the application keystore. Users may now capture a reference to a Signature key ring instead of the alias. This allows implementations to generate the key pair and to rotate the key pairs over time for greater security. Keystore aliases may no longer be used.

The master configuration for Redirection Targets now allows a key ring to be defined instead of the application keystore alias. This provides you with the ability to more easily manage Digital Self Service security.

HTML Sanitization Improvements

The product uses HTML sanitization logic to protect against HTML injection. In this release, the product has improved how it handles custom attributes. Previously, the sanitization logic would automatically sanitize unrecognized custom attributes. Now the sanitization logic has been adjusted to assess the data referenced by the attribute. If the data appears to pose an HTML injection risk, the attribute is sanitized by adding 'data-' to the attribute, rendering it harmless. If the data is considered safe, the custom attribute is accepted as implemented.

Improvements to the HTML sanitization functionality to better handle custom attributes.

Improved Characteristic Type Referential Integrity

The logic to check referential integrity for characteristic types and characteristic values is adjusted to be more targeted, and uses the characteristic entity configuration.

Tables that hold characteristic types and values fall into one of the following categories:

1. Characteristic tables that are a child record of a maintenance object and are used to capture additional information about that object.
2. Log tables that use characteristics to reference additional information related to the log.
3. Configuration tables to define valid characteristics for a related master or transaction table. For example, on the To Do Type, you can configure valid characteristic type for To Do Entries of that type.
4. Configuration tables that use characteristic types for eligibility criteria.
5. Staging tables that capture characteristics as part of an integration step.

Most characteristic tables fall into the first two categories. For these two categories, there is a characteristic entity flag value for each table and the table metadata defines its characteristic entity. For example, the To Do Entry Characteristic table defines the To Do Entry characteristic entity flag value. In this release, those tables have been updated to turn off the Enable Referential Integrity switch for the Characteristic Type and Characteristic Value constraints. Instead, explicit deletion validation logic has been added for Characteristic Type and Value. The logic looks at the Characteristic Entity values associated with the Characteristic Type and will only check the tables related to those values for references to the Characteristic Type/Value.

Tables in category 3 do not reference a characteristic entity value. Since these are configuration tables and low volume, the constraints for Characteristic Type and Characteristic Value continue to turn on the generic referential integrity.

Tables in categories 4 and 5 may or may not have an explicit characteristic entity value depending on the use case. For these types of tables, if there is no Characteristic Entity, it is at the discretion of the product that owns that table as to whether or not referential integrity is enabled for that table.

This enhancement ensures that the deletion logic will not spend time checking every characteristic table for a reference the characteristic type/value. It will first explicitly check the tables related to the characteristic entities configured on the Characteristic Type, and then the standard generic referential integrity based on the constraint configuration will be applied.

Note that as part of this change the referential integrity is also checked when removing a Characteristic Entity value from the Characteristic Type. For example, if you have a Characteristic Type that references the To Do Entry characteristic entity and you want to remove that characteristic entity value, the system will check first that there is no To Do Entry that references that Characteristic Type.

Key Ring Support for OAuth Client and Secret

The product supports securing outbound messages using OAuth authentication. The client ID and secret used to obtain access are stored as message sender context values.

A new OAuth key ring business Object (F1-OAuthKeyRing) allows the client ID and secret to be stored in the same secure repository as other key ring classes, such as Signature keys. The key ring may be configured on the message sender context, replacing the client ID and secret context types. This enables the client secret to be updated periodically without having to update the associated message senders.

In addition, the key ring maintenance object is now an 'owned' entity. Message senders can now be delivered with the appropriate key ring already configured. Users will only have to add their specific keys to the delivered key ring.

New Class Element on Business Flag Standard Name BusinessObject

The F1-BusinessFlagStandardName business object is enhanced to include a new, required element for Class. The class element is mapped to a characteristic of type F1-BSFCL, which has values of Business Flag and Real Time. The description on the **F1-BusinessFlagStandardName** BO has been changed to External Insight Standard Name to reflect its expansion to a wider set of external insights.

Additional class element added to the External Insight Standard Name (formerly described as Business Flag Standard Name) extendable lookup business object allows developers to restrict the choice of standard name types on a business flag type to specific classes.

Support for Capturing and Exporting Batch Run Analytics Data

There is a high volume of batch run data in the operational system that changes frequently and is recorded in a complex set of relationships. This is not well suited for analytics that run directly off the operational data or for being exported for use in other environments. The previous release introduced new tables to support snapshots containing only the batch run and batch thread data applicable to analytics. This release introduces new batch processes designed to populate and export from these tables in increments.

The Batch Run Analytics Snapshot Update and the Batch Thread Analytics Snapshot Update processes are used to add data to the snapshot tables on an ongoing basis. Each snapshot update selects only the batch runs and threads not already in the snapshot. In order to support the initial population of the tables, parameters are provided to limit the records selected to a range of months. Refer to batch controls F1BRANSN and F1BTANSN for more information.

The Batch Run Analytics Snapshot Data Export and the Batch Thread Analytics Data Export processes are used to extract data from the snapshot tables on an ongoing basis. These processes assume that a Data Export Control exists for each of the snapshot maintenance objects to capture the date of the previous extract as a reference point for selecting the next set of records. Refer to batch controls F1BRANEX and F1BTANEX for more information.

The analytics views have not been amended to reference the batch snapshot tables in this release. The batch processes are provided to allow the tables to be populated ahead of the introduction of new batch views in future.

Deprecation Notices for OUAF Version 4.5.0.1.1

This section describes features and system data that are deprecated in this release and planned for deprecation in the future release of Oracle Utilities Application Framework. It contains the following topics:

- [Deprecation in This Release](#)
- [Deprecation Planned for Future Releases](#)

Deprecation in This Release

This section lists the functionalities and system data that are deprecated in Oracle Utilities Application Framework Version 4.5.0.1.1:

- [Support for Migration Requests F1-FrameworkAdmin and F1-SchemaAdmin](#)
- [Work Calendar Legacy Page Metadata](#)
- [Time Zone Legacy Page Metadata](#)
- [Application Viewer](#)
- [Invite User to Mobile Application Zone](#)
- [Mobile Remote Message Artifacts](#)
- [Mobile Data Terminal Artifacts](#)
- [Mobile Component Artifacts](#)
- [Mobile Data Terminal Type Artifacts](#)
- [Deployment Part Artifacts](#)
- [Deployment Type Artifacts](#)
- [Deployment Artifacts](#)
- [Master Configuration Business Objects Related to Mobile Application Framework](#)
- [Migration Plan - Import Algorithms](#)
- [User Group Service Management Portal](#)

Support for Migration Requests F1-FrameworkAdmin and F1-SchemaAdmin

These two migration requests have been marked as not recommended for a long time. As of this release, they are no longer delivered in base.

For backward compatibility, the two migration requests are updated to have a "CM" owner for upgrading utilities. To avoid confusion, you may remove the records from your installation if they have never been used.

Work Calendar Legacy Page Metadata

The following navigation keys have been removed:

- workCalendarMaint
- workCalendarMainPage
- workCalendarHolidayGrid
- workCalendarMainPage_H

Records in the UI metadata program component tables related to the above navigation keys.

Time Zone Legacy Page Metadata

These navigation keys have been deprecated:

- timeZoneMainPage
- timeZoneMainPage_H
- timeZoneTabMenu

Records in the UI metadata program component tables related to the above navigation keys.

Application Viewer

Note that there is currently no replacement for the Javadocs viewer. This is planned to be addressed in the next release.

Invite User to Mobile Application Zone

This zone was implemented as part of the mobile application, which is no longer supported.

Mobile Remote Message Artifacts

The user interface metadata has been removed for this object. This table was originally configured to support ILM, but it is no longer included in ILM partitioning steps.

Batch controls related to processing this table were removed if no instance data in the Batch Run table is found. If instance data is found, the batch control metadata is retained with a CM owner flag.

Mobile Data Terminal Artifacts

The user interface metadata has been deprecated for this object.

Mobile Component Artifacts

The user interface metadata and migration plan pan have been removed for this object.

Mobile Data Terminal Type Artifacts

The user interface metadata and migration plan have been deprecated for this object.

Deployment Part Artifacts

The user interface metadata and migration plan have been removed for this object.

Deployment Type Artifacts

The user interface metadata and migration plan have been deprecated for this object.

Deployment Artifacts

The user interface metadata has been removed for this object.

Master Configuration Business Objects Related to Mobile ApplicationFramework

The master configuration business objects F1-MobileConfigurations and F1-MobileIdentityConfiguration are no longer delivered in base. Artifacts related to these BOs are removed if no master configuration records are found for the respective BOs. If there is a master configuration found for either, then the related data remains with a 'CM' owner.

The REST service and corresponding script to update the master configuration is deprecated.

- IWS: F1-MobileIdentityConfiguration
- Script: F1-MobIdMCfg

Migration Plan - Import Algorithms

The Import algorithm plug-in spot for migration plan was flagged for deprecation many years ago and the base product does not deliver algorithms for this plug-in spot. The Pre- Compare plug-in spot is the recommended mechanism for adjusting migration data during import.

User Group Service Management Portal

This functionality was moved entirely from the **User Group Service Management** portal to the **Service Manager** tab on the new **User Group** portal.

Deprecation Planned for Future Releases

This section lists the functionalities and system data that will be deprecated in the future releases of Oracle Utilities Application Framework:

- [Support for Previous User Experience](#)
- [F1-DFLTAPS and FWLZDEXP Application Services](#)
- [Workflow and Notification Metadata and Database Tables](#)
- [Mobile Application Framework Metadata](#)
- [Key Ring Validation Scripts, Algorithm Types, and Algorithms](#)
- [UI Metadata Related to Converted Pages](#)
- [Miscellaneous System Data](#)
- [XSLT Managed Content Type](#)
- [REST IWS - Original REST Servlet](#)
- [Append Setting from Pagination](#)
- [Support for Master/Subordinate Servers for Web Service Catalog](#)
- [Batch Run Statistics Portal Functionality](#)
- [Configuration Migration Assistant Import Algorithm Plugin Spot](#)
- [F1-MAINPROC Business Object Read When Pre-processing Exists](#)

Support for Previous User Experience

In the current release, the product provides an option to Switch UI View. This changes the current user experience for the user for that session. Assuming the user is in the latest user experience (referred to as Redwood), this action switches the user experience to the previous look-and-feel.

In the future, the product plans to remove support for the ability to switch that design that preceded Redwood.

F1-DFLTAPS and FWLZDEXP Application Services

In an effort to consolidate application services, the product is removing all references in base delivered metadata to F1-DFLTAPS and FWLZDEXP. Records will instead reference F1-DFLTS.

You should select **Admin → Security → Application Service** and view F1-DFLTAPS. Review the Secured Objects zone. Note especially if your implementation uses this application service within HTML or Schemas. Any references should be adjusted to F1-DFLTS or a different application service that is appropriate for the business rule.

Implementations should repeat the above steps for the application service FWLZDEXP. In addition, if your implementation has any hard-coding of either F1-DFLTAPS and FWLZDEXP in code that is not detectable by the Secured Objects zone, those references should also be updated to use F1-DFLTS or a different application service that is appropriate for the business rule.

Workflow and Notification Metadata and Database Tables

Workflow and notification functionality was an early way to support exchanging messages with an external system (notification) and providing a configurable process for acting on incoming messages (workflow). In more recent years, the functionality for managing external messages is supported using Outbound Message and Inbound Web Service functionality. In addition, there are several features to support processing incoming messages. Service scripts can handle simple use cases. For more complicated processes, the service task or other business object driven objects are available.

The metadata and database tables related to this feature will be removed in a future release. Note that only a portion of the functionality for this feature is managed by Oracle Utilities Application Framework.

Mobile Application Framework Metadata

Removal of support for the Mobile Application Framework has already been announced in a previous release. However, there is metadata still included in the application related to this functionality.

The metadata will be removed in a future release.

Key Ring Validation Scripts, Algorithm Types, and Algorithms

The product is removing all scripts, algorithm types, and algorithms that performed validation rules on the K1-SignatureKeyRing business object. The algorithms have been removed from the BO configuration. There are requirements to expand the use of a signature key ring beyond the current implementation for object file storage and the existing validations are not applicable to other planned use cases.

The following items will be removed in a future release.

- Algorithm:
 - K1-KRDCKFS
 - K1-KRINCKFS
- Algorithm Type:
 - KRDCKFS
 - K1-KRINCKFS
- Message:
 - 11009 / 1402
- Plugin Script:
 - K1-KRDCKFS
 - K1-KRINCKFS
- Service Script:
 - K1-ChkCfgExL

UI Metadata Related to Converted Pages

The UI metadata related to fixed pages that have been converted to portals will be removed in a future release. The navigation keys listed are related to each maintenance page. The related UI program component data will also be removed. Note that the metadata related to the search pages will not be removed at this time in case they are used on other fixed pages.

- To Do Entry Maintenance
 - toDoEntryCharGrid
 - toDoEntryDrillKeyValuesListGrd
 - toDoEntrySortKeyValuesListGrid
 - todoentrykeyvalue
 - todoentrymain
 - toDoEntryMaint
 - toDoEntryPopupAdd
 - toDoEntryPopupForward
 - toDoEntryPopupSendBack
 - Any help navigation keys
- Table Maintenance
 - metaDataTableFieldsGrid
 - metaDataTableMainPage
 - metaDataTableCFldsGrid
 - metaDataTableConstPage
 - metaDataTableMaint
 - metaDataTableRefByConstPage
 - metaDataTableFieldPage
 - Any help navigation keys
- Work Calendar Maintenance
 - workCalendarMaint
 - workCalendarMainPage
 - workCalendarHolidayGrid
 - Any help navigation keys
- Message Maintenance
 - msgMaintDetailsPage
 - msgMaintGrid
 - msgMaintPage

- msgMaintTabMenu
- Any help navigation keys
- Time Zone Maintenance
 - timeZoneMainPage
 - timeZoneTabMenu
 - Any help navigation keys
- Application Security Portal
 - f1appsecTabMenu
- Display Icon Portal
 - displayIconRefMaint

Miscellaneous System Data

The following miscellaneous data will be deprecated in the future release:

Object	Data	Description/Comments
Lookup Value	CHAR_ENTITY_FLG/F1SE	Characteristic Entity/Sync Request Inbound Exception
Script	F1-TDMgActSS	To Do Management – Process Actions (Deprecated) / Replaced by F1TDMgActSS
Script	F1AddDebugLg	Add Log for Monitoring Probe (Deprecated) / Replaced by a BS - F1-MONPRBLOG
Zone	F1-BOMOSRCH	Not in use by base functionality
Zone	F1-CATCHSCH	Not in use by base functionality
Zone	F1-MONAVKEY	Not in use by base functionality
Zone	F1-REVCONQRY	Not in use by base functionality

XSLT Managed Content Type

Entries in the Managed Content table related to XSL should be using the XSLTC managed content type and not the XSLT managed content type. In a future release, the XSLT managed content type will no longer be supported.

REST IWS - Original REST Servlet

The original URL supplied for invoking IWS based REST services included the IWS Service name in its makeup. Support for this will continue for backward compatibility purposes, but it will be deprecated in a future release. You should adjust your existing integrations to use the currently supported URL.

Append Setting from Pagination

There are several known issues with the functionality of the "append" option in pagination. It is recommended that you do not use this pagination setting.

Support for Master/Subordinate Servers for Web Service Catalog

The Service Catalog Configuration (master configuration) enables you to define subordinate servers. Defining subordinate servers is no longer applicable for the Oracle Integration Cloud.

Batch Run Statistics Portal Functionality

The **Batch Run Statistics** portal provides additional information about batch runs, but some functionality on the portal is related to capturing additional information from an external tool. This information is stored in a Fact record. Support for capturing additional information from an external tool will be discontinued in a future release.

Configuration Migration Assistant Import Algorithm Plugin Spot

The Content Migration Assistant Import algorithm plug-in spot will be deprecated. It is recommended that you review any existing algorithms and create appropriate Pre- Compare algorithms instead.

F1-MAINPROC Business Object Read When Pre-processing Exists

In the original implementation of configuration tools, the main framework maintenance BPA (F1-MainProc) did not perform a Read of the BO when a pre-processing script was linked to the BO via options. The pre-processing script was responsible for the Read.

In a subsequent release, a BO Read was added in F1-MainProc (even if a pre-processing script existed) to resolve a UI Hint issue related to child business objects. This solution introduced a problem only visible for specific scenarios and a different fix has been introduced. The new fix made the BO Read unnecessary in F1-MainProc. Because there are many pre-processing scripts that are properly performing the Read of the BO, ideally the BO Read should be removed from F1-MainProc so that multiple reads are not performed. However, there may have been pre-processing scripts introduced after the BO Read was included in F1-MainProc that were coded to not perform a BO read in the pre-processing script. Due to this situation, the BO Read is still performed as part of the processing of F1-MainProc.

When a pre-processing script exists, we plan to remove the BO Read from F1-MainProc logic. You should review your custom pre-processing scripts that are linked to your BO options to ensure that they properly perform a Read of your BO.

Deprecation Notices for ORMB Version 6.1.0.0.0

This section describes features and system data that are deprecated in this release and planned for deprecation in the future release of Oracle Revenue Management and Billing. It contains the following topics:

- [Deprecation in This Release](#)
- [Deprecated Platforms](#)
- [Deprecation Planned for Future Releases](#)

Deprecation in This Release

The following functionalities and system data are deprecated in this release of Oracle Revenue Management and Billing:

- [Support for @INSTALL_DIR](#)

Support for @INSATLL_DIR

In the On Premise and Cloud Service application environments, the `@INSTALL_DIR` variable is no longer supported. You cannot upload, download, extract, or read the files from the installed directory on the server.

Deprecated Platforms

As per Oracle revised guidelines, upcoming releases of ORMB will not be certified on IBM Websphere application server post March 2025.

Deprecation Planned for Future Releases

The following table lists the objects which will be deprecated in the future release of Oracle Revenue Management and Billing:

Object Type	Object Name
Algorithm Type	C1_CURALG, SA_DERV_POPC, C1-AUDEVMPPR, C1-PLASGNAU, C1-PRCASGNAU, C1-PLAUALG
Algorithm	C1-AUDEVMPPR, C1-PLASGNAU, C1-PRCASGNAU, C1-PLAUALG
Feature Configuration	C1_EX_ROUND
Option Types	Currency Conversion Algorithm and Payment Distribution To-Do (from the C1_MLTCURACC feature configuration)

Object Type	Object Name				
Table Columns	<p>The following table lists the columns which will be deprecated in the next release:</p> <table border="1"> <thead> <tr> <th>Table Name</th> <th>Column Name</th> </tr> </thead> <tbody> <tr> <td>CI_ACCT_PER</td> <td>BILL_RTE_TYPE_CD, RECEIVE_COPY_SW, BILL_FORMAT_FLG, NBR_BILL_COPIES, CUST_PO_ID, NOTIFY_SW, and BILL_ADDR_SRCE_FLG</td> </tr> </tbody> </table>	Table Name	Column Name	CI_ACCT_PER	BILL_RTE_TYPE_CD, RECEIVE_COPY_SW, BILL_FORMAT_FLG, NBR_BILL_COPIES, CUST_PO_ID, NOTIFY_SW, and BILL_ADDR_SRCE_FLG
Table Name	Column Name				
CI_ACCT_PER	BILL_RTE_TYPE_CD, RECEIVE_COPY_SW, BILL_FORMAT_FLG, NBR_BILL_COPIES, CUST_PO_ID, NOTIFY_SW, and BILL_ADDR_SRCE_FLG				
Batch Control	GLASSIGN, C1-IAENT, C1-DARSU, BILLING				
View	CI_EFF_ACCT_PRICING_VW, CI_EFF_PER_PRICING_VW				
Business Service	<p>C1-EffectivePricing, C1_PriceParmBS</p> <p>Note: Instead of using the C1-EffectivePricing business service, use the C1-GetEffectivePricing business service to view the data on the Pricing (Account) and Pricing (Person) screens. Similarly, instead of using the C1_PriceParmBS business service, use the C1_PRICE_PARM business object to add, edit, copy, and delete a parameter.</p>				
Service Program	EFFPRCSERVICE, C1_PRICEPARM				
Column	ADDRESS1, ADDRESS2, ADDRESS3, ADDRESS4, CITY, NUM1, NUM2, COUNTY, POSTAL, HOUSE_TYPE, GEO_CODE, IN_CITY_LIMIT, STATE, and COUNTRY from the CI_PER table				
Table	CI_PER_ADDR_SEAS				
Screen	Rate Check, Policy (P&C), Insurance Control Central, Collection Control Central, Account Current, Variance Parameter				

Therefore, we strongly recommend you not to use these objects in any custom implementation.

Product Documentation

User manuals and other technical documents are available in the Portable Document Format (PDF) format. You can download the release-specific documentation from either of the following locations:

- **Oracle Technology Network (OTN)** – You can access the ORMB release-specific documentation libraries from OTN using the following URL:

<http://www.oracle.com/technetwork/indexes/documentation/fsgbu-1364781.html>

It contains the **Documentation Library for Oracle Revenue Management and Billing On-Premise Solution** section. You can view and download a release-specific documentation library by clicking the **View Library** and **Download** links, respectively, corresponding to the respective ORMB version.

- **Oracle Help Center (OHC)** – You can access the ORMB release-specific documentation from OHC using the following URL:

<https://docs.oracle.com/en/industries/financial-services/revenue-management-billing/index.html>

The ORMB OHC page contains a drop-down list which allows you to select the ORMB version for which you want to access the documentation.

Points to Note:

You can access the documentation of a release prior to 5.0.0.0.0 from OHC by selecting the **Previous Releases** option from the list.

Always ensure that you download latest revision of the document from **OTN** or **OHC**.

From 6.0.0.0.0 release onwards, the Oracle Revenue Management and Billing Online Help (i.e., help.ear file) is not packaged with the application. The ORMB Online Help is published on Oracle Help Center (OHC). You can continue to use context-sensitive help from the application which will point to the respective topic on OHC.

Documentation Updates

Here are some updates with respect to the ORMB documentation:

- From 6.0.0.0.0 release onwards, the Oracle Revenue Management and Billing Online Help (i.e., help.ear file) is not packaged with the application. The ORMB Online Help is published on Oracle Help Center (OHC). You can continue to use context-sensitive help from the application which will point to the respective topic on OHC.
- You can also directly access the online help on OHC in the standalone mode using the following URL:
https://docs.oracle.com/en/industries/financial-services/revenue-management-billing/61000/ormb-online-help/Topics/ORMB_Intro.html
- The Reporting User Guide is decommissioned and no longer supported. For information related to the reporting feature, you need to refer the **Reporting** chapter in the Banking User Manual or Insurance User Manual.
- Oracle Utilities Application Framework Version 4.5.0.1.1 Software Development Kit Guide is available along with the release-specific documentation on the ORMB OHC page.


Media Pack Download

Oracle Financial Services Revenue Management and Billing 6.1.0.0.0 and Oracle Insurance Revenue Management and Billing 6.1.0.0.0 media packs can be installed on the following supported platforms:

- AIX (64-bit)
- Microsoft Windows (64-bit)
- Linux (64-bit)

The media pack includes multiple packages. For more information, refer to the **Media Pack Contents** section in the *Oracle Revenue Management and Billing Version 6.1.0.0.0 Quick Installation Guide*.

To download the media packs:

1. Login to [Oracle Software Delivery Cloud](#).
2. Select the **Release** option from the **All Categories** list.
3. Type **Oracle Financial Services Revenue Management and Billing** or **Oracle Insurance Revenue Management and Billing** in the corresponding text box.
4. Click **Search**. A list of media packs appears in the search results.
5. Click the **Select** () icon corresponding to the **Oracle Financial Services Revenue Management and Billing 6.1.0.0.0** or **Oracle Insurance Revenue Management and Billing 6.1.0.0.0** media pack.
6. If required, you can verify the media pack which is selected by clicking the **View Items** link.
7. Click the **Continue** link. A page appears.
8. Select the required platform from the **Platforms/Languages** list.
9. Click **Continue**. A page appears with the license agreement.
10. Scroll and read the entire license agreement and then click the **I reviewed and accept the Oracle License Agreement** option.
11. Click **Continue**. The **File Download** page appears.
12. Ensure that all the packages in the media pack are selected and then click **Download**. An executable file is downloaded on your local machine.
13. Run the executable file. The **Oracle Download Manager** window appears.
14. Browse to the location where you want to download the packages and then click **Next**. The packages are downloaded on your local machine.

Bug Fixes

The following table lists the bugs that are fixed in this release:

Bug Number	Copy of (Base Bug)	Description
36085419	36058178	BILL VS OPEN ZONE DID NOT SHOW ANY RECORD
36060622	35968385	TENDER CONTROL UI DOES NOT DISPLAY TENDERS LIST BY DEFAULT
36026131	35960956	JAVA ERROR WHILE RUNNING THE FRT CM_PERACCT_T1
36026130	35922396	ALL TRIAL BILL SEGMENTS ARE NOT DISPLAYED ON TRIAL BILL UI WHEN USER NAVIGATE BACK FROM OTHER SCREEN
36026128	35929839	DEAL: CBPR PAIR PRODUCT IS NOT VISIBLE IN BESPOKE PRICELIST
36012314	35765762	RMB V5.1 - TRIAL BILLING ISSUE/OBSERVATION FOR MANUAL PROCESS
36001966	35960449	TENDER DETAILS NOT GETTING UPDATED TO INDIVIDUAL ACCOUNT DETAILS WHEN PAYMENT IS TRANSFERRED FROM SUSPENSE ACCOUNT-FORWARDPORT
35997180	35835735	DEAL: ERROR ON CLICK ON SIMULATE - SERVICE QUANTITY NOT FOUND IN SQ COLLECTION(„TXN_VOL)
35987935	35940876	FW PORT : REPROCESS CNCL MEMBERSHIP ISSUE
35972394	35680788	TRANSACTION CLEANUP BATCH IS RUNNING FOR LONG TIME
35971625	35580842	PRODUCT VERSION IS NOT DUPLICATED WHEN CHECK NEXT CONDITION EXISTS
35971546	35595697	DBMS IMPORT IS NOT WORKING IN CFS ENVIRONMENT
35971494	35891874	DEAL MANAGEMENT: CORRECT THE LABEL - SELECT PRICE ITEM GROUP SELECTION TO SELECT PRICE ITEM GROUP
35971206	35810329	SPREAD PERCENTAGE AT PRICE ASSIGNMENT LEVEL FAILED WHEN DEAL CREATED AT CUSTOMER LEVEL.
35962321	-	DEPRECATED UI
35961909	35947998	PARENT-CONSTRUCT HIERARCHY MOVEMENT
35961907	35801023	SYSTEM GIVING IMPROPER ERROR WHILE UPLOADING ODB FILE WITH DECIMAL FOR JPY CURRENCY
35961906	35890036	DEAL: PRODUCT DELETED FROM DEAL IS STILL AVAILABLE IN BESPOKE PL BUT NOT SHOWING
35947959	35930453	EXCEPTION THROWN WHILE SELECTING STATE AT REFUND

Bug Number	Copy of (Base Bug)	Description
35945009	35870568	SOME VALIDATIONS MISSING AND DEFECTS IN BILL LEVEL UPLOAD FUNCTIONALITY - FORWARDPORT
35927370	35799152	PRICING GROUP RULE BILL LEVEL TABLE
35927229	35903040	APTC BILLABLE CHARGES NOT GETTING CANCELLED WHEN APTC AMOUNT UPDATED TO \$0.00
35919033	35846222	ERROR IN UPCOMING BATCH JOB SCHEDULER IN SCHEDULER JOB MONITOR DASHBOARD
35919032	35810677	DEAL : DEAL LOGS LIMIT SHOULD BE INCREASED TO 4000 CHARACTERS
35919031	35732038	TRANSACTION CLEANUP BATCH IS RUNNING FOR LONG TIME.
35892412	-	MEMBERSHIP REPRICING WEB SERVICES
35866570	35784979	BILL DISPLAY BUTTON IN CUSTOMER 360 THROWING A SERVER ERROR ON CLICK
35854259	35805336	PRIMARY IDENTIFIER SEARCH NOT SUPPORTING CASE-INSENSITIVE
35854258	35789458	USAGE ACCOUNT OF ONE ACCESS GROUP ARE VISIBLE TO USER IN OTHER ACCESS GROUPS IN CUSTOMER 360 VIEW
35850319	35600892	FT SERVICE ENHANCEMENTS
35835576	-	HOLD REPRICING
35814953	35814945	FORWARDPORT - DATA EXTRACT BASE BATCH C1-DTEX IS NOT ABLE TO EXTRACT DATA USING MEMBERSHIP (C1-MEMBER) MO
35798998	35449589	DEAL MANAGEMENT: ROUNDING ISSUE
35796496	35566880	BILLABLE CHARGES CALCULATED INCORRECTLY FOR PRORATION RULE TYPE AS MDMT
35675812	35376889	PAYMENT NOT GETTING APPLIED PROPERLY AFTER OVERPAYMENT DISTRIBUTION - FWP V61
35584877	35384433	THIS ENTRY EXISTS ON TABLE FINANCIAL TRANSACTION - FP61000
28500558	-	FULLY INSURED CUSTOMER LEVEL DELINQUENCY & GAPS
36080003	36002150	DEAL:UNABLE TO DISPLAY PRICING IN A TEXTBOX MORE THAN 2 DIGITS
36079860	36002150	DEAL: UNABLE TO APPROVE THE DEAL BECAUSE FEW PRICEITEMS ARE NOT HAVING ANY APPROVE/UNAPPROVED STATUS
36057814	35866762	ABILITY TO ENABLE OR DISABLE EXCEL UPLOAD AND DOWNLOAD FEATURE IN DEAL

Bug Number	Copy of (Base Bug)	Description
36036415	35775982	CONCERN WITH STEPS TO DELINK AND CORRECT PARENT ID
36036272	35626203	CONCERN WITH STEPS TO DELINK AND CORRECT PARENT ID
36030726	35068706	BILLS CREATED THROUGH C1-BLCRN BATCH WITH POST PROCESSING CHARGES ARE NOT GETTING CANCELLED
35974471	35777472	TRANSFER ADJUSTMENT IS NOT CONSISTENT WHEN WE CLICK ON SAVE AND GENERATE
35972145	35875369	CUSTOMER 360 CHILD PERSONS ZONE PAGINATION DISPLAYS INCORRECT RESULTS AND CONTAINS DUPLICATE RECORDS -- FORWARDPORT
35970860	35895920	HEALTH PLAN PRICING RULE REPRICING REQUEST ISSUE -- FORWARDPORT
35970251	35579198	BILLABLE CHARGES ARE NOT GETTING CANCELLED IN THE CASE OF REPRICING CHAR DELETION
35961961	35720291	PATCH 35624175 INSTALLED. BILLCOMPLETIONSETTLEMENTSTAMPINGALG_IMPL -QUERY ISSUE
35950144	35626077	REFUND REQUEST CREATED SINGLE MATCH EVENT FOR MULTIPLE BILL SEGMENT - FORWARDPPORT
35937470	35500190	PAYOR ACCOUNT MISSING WHEN DOING PAYMENT TRANSFER
35916776	35788449	CURRENTBG OPTION RESULTING IN INCORRECT SL ACCUMULATION WHEN MEMBER MOVES ACROSS BILLING GROUP
35903061	35454236	DEAL: CBPR PRICE CHANGE IMMEDIATELY FOR ALL PAIRED PRODUCTS (ON TAB OUT)
35877431	35793507	NOT ABLE TO REPLACE FILE ON BILLABLE CHARGE UPLOAD SCREEN
35858781	35288144	HEALTH PLAN PRICING UI ZONE-FORWARDPORT - 6.1.0.0.0
35855704	35783516	MEMBERSHIP INFO ALGORITHM IS FAILING WITH NULL POINTER EXCEPTION WHEN WE TRY TO ADD A PRODUCT - FORDWARDPORT
35854778	35722444	NEED SOLUTION TO PREVENT DEAGG REQUESTS FROM BEING SUBMITTED IN MIDDLE OF DEAGG PROCESS
35845087	35709794	QUERIES IDENTIFIED IN APAYCRET - FORWARDPORT
35814277	35741713	LINK FOR TRANSACTION DETAILS IS FAILING AFTER 1ST TRANSACTION DETAILS IS EXPANDED
35805538	35700135	ISB NULL POINTER EXCEPTION RATECOMPONENTVALUEFORHEALTHCARE

Bug Number	Copy of (Base Bug)	Description
35805273	35706778	POST INSTALLATION OF PATCH 35635889 - DIFFERENCE IN THE AMOUNT -- FORWARDPORT
35805205	35673313	ORIGINAL END DATE ON THE MEMBERSHIP -- FORWARDPORT
35803034	35579955	REPRICING RECORDS ARE NOT GETTING CREATED IF WE UPDATE-ADD NEW CHAR ON MEMBERSHIP PERSON
35803030	35571512	REPRICING ENTRIES ARE NOT CREATED FOR FEW CHARS
35803023	35759439	DOB CHANGE FOR A PERSON AND NEW MEMBER IN SAME INBOUND MESSAGE- INCORRECT BILLABLE CHARGE
35802974	35656377	ISSUE PERSIST, AFTER INSTALL PATCH 35584162.HILD BILLABLE CHARGES ARE NOT GETTING CANCELLED WHEN THERE IS CHANGE IN PRIMARY PRICING RULE
35785833	35753069	TERMINATION ISSUE CAUSED BECAUSE OF THE RENEWAL DATE -- FORWARDPORT
35785798	35752513	ISB INSERT DATA INTO CI_REPRC_REQ_DTL TABLE HAS SELECT ON DUAL -- FORWARDPORT
35785768	35747248	HEAVY CONCURRENCY WAITS , INDEX CONTENTIONS & UNOPTIMIZED READS OBSERVED -- FORWARDPORT
35784295	35700135	ISB NULL POINTER EXCEPTION RATECOMPONENTVALUEFORHEALTHCARE
35777016	35306723	PREMIUM AND BENEFIT SHOULD CALCULATE AS PER THE NUMBER OF DAYS(PRORATED BASIS)
35772890	35588259	EDITING SSL AND ASL PRICING RULE IS THROWING ERROR WHEN USING LINE ITEMS
35736442	35626697	ORMB 2.8 FREEZING THE BILL SEGMENT IS THROWING UNIQUE CONSTRAINT VIOLATION ERROR
35735489	35653731	REPRICING ENTITIES ARE GETTING CREATED BEYOND MEMBERSHIP END DATE --FORWARDPORT
35735475	35677247	C1-FIBCR GETTING IN TO ERROR WHEN INITIAL MEMBERSHIP GET ERRORS DUE TO PRICING NOT FOUND AND IT WAS-- FORWARDPORT
35735466	35689078	ISB QUERIES WITH TRIM IDENTIFIED IN DATABASE-- FORWARDPORT
35735454	35678379	INDIVIDUAL MEMBERSHIP C1-REPC1 ERROR WITH ACCOUNT ID TYPE/VALUE AUDIT EVENT--FORWARDPORT
35727508	35531523	COMPLETED PAY INSTRUCTION IS NOT REOPENED AFTER A TAKEBACK PAY INSTRUCTION FOR IT HAS BEEN CANCELLED - FORWARDPORT

Bug Number	Copy of (Base Bug)	Description
35727454	35577920	DBA OBSERVATION ORMB PROD MODULE C1-RRADJ SQL 1BYWPUTTKGPH9 - FORWARDPORT
35727433	35635889	PRORATION UI FOR PRORATION FACTOR AND AMOUNT - FORWARDPORT
35727411	35498826	C1-SEGWRAPSVC & C1-FTSERVICE - THESE WRAPPER SERVICES FOR BILL SEGMENT & FT ARE EXPERIENCING AN ISSUE - FORWARDPORT
35696847	35600892	FT ENTITY CREATION USING C1-FTSERVICE BASE SERVICE , MISSING ENTRY IN C1_FT_EXT, C1_FT_EXT_CHAR.
35696544	35589848	HEALTHPLAN - HIOS UPDATE
35694462	35678240	PRIMARY IDENTIFIER SEARCH NOT SUPPORTING CASE-INSENSITIVE WORDS-FORWARD PORT
35653944	35585679	RATE GUARANTEE PRICING RULE ID IS NOT GETTING DERIVED CORRECTLY
35644260	35590620	COVERAGE DATES NULL ISSUE FOR PLAN ADD SCENARIO -- FORWARDPORT
35630133	35584162	HILD BILLABLE CHARGES ARE NOT GETTING CANCELLED WHEN THERE IS CHANGE IN PRIMARY PRICING RULE --FORWARDPORT
35570356	35403336	BILL INFORMATION PORTAL ISSUE NOT INCLUDING BILL IN PENDING/CANCELLED FT/BILL SEGMENT STATUS
35554838	35369544	REVIEW AND CONFIRM IF SPECIFIC PATCH'S(4.0 VERSION) HAVE BEEN FORWARD PORTED TO ORMB 5.1
35536209	35469549	REPRICE MEMBERSHIP CHAR TABLE POPULATION FAILING WITH UNIQUE CONSTRAINT ERROR-FORWARDPORT
35536185	35419232	HIBERNATE QUERY ERROR, PATH EXPRESSION ENDS IN COMPOSITE VALUE IS THROWN WHEN GETTING EFFECTIVE CHAR - FORWARDPORT
35536180	35355398	HEALTH CARE INBOUND MESSAGE MONITOR - THREADING ISSUE - FORWARDPORT
35525889	35387079	SYSTEM ERROR WAS SHOWING WHEN EDITING THE DISCOUNT SHARE PRICING
35525651	35373642	BILLABLE CHARGES ARE NOT GETTING CANCELLED-FORWARDPORT
35512755	35477947	FIXES FOR PATCHES 33170313 AND 33363241 TO ORMB 4.0 - FORWARDPORT
35508487	35482472	MEMBERSHIP PERSONS ZONE NOT DISPLAYING ALL ACTIVE MEMBERSHIPS

Bug Number	Copy of (Base Bug)	Description
35438253	35378084	ADDITIONAL REPRICING ENTITIES ARE GETTING CREATED INCASE OF MEMBERSHIP UPDATE
35419005	35387044	ORMB VERSION 5.1 - UI ISSUES - FORWARDPORT
35418966	35350026	NEWBORN WAIVER IS NOT GIVEN TO NEWBORN CHILD - FORWARDPORT

Known Issues

This section lists the known issues in Oracle Revenue Management and Billing Version 6.1.0.0.0 along with workarounds available to handle these issues. The known issues are grouped into the following categories:

- [Framework](#)
- [Banking](#)
- [Insurance](#)
- [Documentation](#)

Framework

Issue	GETTING EXCEPTION ERROR MESSAGE ON "CONVERSION ENTITY" UI.
Description	While viewing the details of a conversion entity, an exception error occurs in the Duplicate Keys zone of the Conversion Entity Dashboard screen.
Workaround	None

Issue	SERVER ERROR OCCURRING AFTER DELETING A "MIGRATION REQUEST".
Description	While deleting a migration request, a server error appears in the right panel even though the migration request is deleted from the system.
Workaround	None

Issue	COLOR CONTRAST DOES NOT MEET ACCESSIBILITY REQUIREMENTS
Description	At present, the application color contrast does not meet the accessibility guidelines.
Workaround	None

Issue	GETTING EXCEPTION ERROR MESSAGE ON "MESSAGE" UI.
Description	While searching using the percentage (%) symbol in the numeric fields (i.e. Message Category From and Message Category To), an exception error occurs instead of showing an appropriate error message in the Message Search zone.
Workaround	None

Banking

Issue	LOCALIZATION ISSUE WHEN MAKER AND CHECKER CHOOSE DIFFERENT LANGUAGES
Description	When a maker and checker operate in different languages, a maker cannot see the approved transactions.
Workaround	Maker should login with a language that checker is using to see the approved transactions.

Issue	FOREIGN KEY AND FILE LOCATION CHARACTERISTIC TYPE NOT SUPPORTED IN UI MAPS
Description	The screens created using the UI map do not support the Foreign Key Value and File Location Value characteristic types. These screens only support the Adhoc Value and Predefined Value characteristic types.
Workaround	None

Issue	MESSAGE DOES NOT APPEAR ON CLICKING BUTTONS IN LIST OF PRICE ASSIGNMENTS ZONE
Description	On approving, rejecting, or canceling one or more price assignment requests (at once) from the List of Price Assignments zone, the appropriate message does not appear when you click the Accept Changes , Return to Submitter , or the Revert to Original button.
Workaround	None

Issue	C1-TXCNC BATCH GETS EXECUTED SUCCESSFULLY EVEN IF C1-TXNCU BATCH FAILS
Description	<p>When you execute the C1-TXNCU batch with either of the following parameters during the cancellation process, an error occurs:</p> <ul style="list-style-type: none"> • Transaction Source • Division <p>If you further execute the C1-TXCNC batch during the cancellation process, the status of all transactions in the feed is changed to Cancelled (CNCL). But, in this case, the SQIs on the billable charges are not accurate.</p>
Workaround	None

Issue	CONDITIONAL APPROVAL WORKFLOW CANNOT BE USED WHILE COPYING A PRICE LIST
Description	The system allows you to define conditional approval workflow for business objects. However, at present the conditional approval workflow cannot be used while copying a price list.
Workaround	None

Issue	BILLS GENERATED FOR THE MEMBER ACCOUNT AND NOT FOR THE MASTER ACCOUNT
Description	If you add a contract to a member account after the account is added to the master account, the system will not duplicate the newly added contract at the master level. Therefore, in such scenarios, billable charges related to the newly added contract will be billed to the member account and not to the master account.
Workaround	You need to ensure that no new contracts are added to the member account after the account is added to the master account.

Issue	INCORRECT RESULTS IF DISAGGREGATION BATCHES NOT EXECUTED IN SEQUENCE
Description	<p>During the transaction disaggregation process, you must execute the following batches in the specified order:</p> <ol style="list-style-type: none"> 1. Identify Affected Transactions (C1-IAENT) 2. Process Non Aggregated Transactions (C1-PDTXN) 3. Clean Up (C1-TXNCU) 4. Update Disaggregation Request Status (C1-DARSU) <p>Otherwise, erroneous results might occur.</p>
Workaround	None

Issue	TWO CONCURRENT RUNS WHICH DERIVE SAME DIVISION FOR TRANSACTIONS DOES NOT WORK
Description	If you execute a batch concurrently with two different divisions (for example, D1 and D2), the erroneous results might occur when transactions in both the runs derive the same division, account, and/or product combination.
Workaround	None

Issue	ERROR OCCURS WHEN YOU DISPLAY BILL FOR A PRODUCT THAT BELONGS TWO RELATIONSHIPS
Description	If a product is added to two or more product to product relationships using the relationship type as Service, the system error occurs when you display bill for the product using Documaker. For example, if P3 is added in the P1 and P2 product relationship using the relationship type as Service, an error occurs when you display bill for P3 using Documaker.
Workaround	None

Issue	ACTIVE CONSTRUCTS DETERMINED ON CUT-OFF DATE INSTEAD OF CHARGES START & END DATE
Description	The system determines the active construct based on the cut-off date and bills the usage accounts' charges through an invoice account which is defined in the active construct. It does not determine the active construct based on the billable charge's start and end dates.
Workaround	None

Issue	BILL SEGMENTS CREATED BUT FTS NOT CREATED WHEN ACCOUNTING CALENDAR NOT DEFINED
Description	There might be situations when you generate bill segments for a pending bill which is created for a bill cycle whose accounting date either falls within the closed accounting calendar or does not fall within any accounting calendar. In such scenarios, the bill segments are generated, but the financial transactions are not created for the bill segments. An error occurs when you view such bills, whose financial transactions are not created, through the Bill screen.
Workaround	None

Issue	STATUS OF ALL LEGS CHANGED TO ERROR IF EXCHANGE RATE NOT AVAILABLE FOR ONE LEG
Description	If a transaction has multiple legs and the system could not find exchange rate for one of the leg while executing the C1-TXNSQ batch, the status of the transaction and all its transaction legs is changed to Error .
Workaround	None

Issue	INCORRECT RESULTS APPEAR WHEN UNDERSCORE CHARACTER IS USED IN SEARCH CRITERIA
Description	If you use the underscore (_) character in the search criteria, the system does not search strings with the underscore character. Instead, the system interprets the underscore (_) character as a wildcard character.
Workaround	None

Issue	PAYMENT AMOUNT IS INCORRECT WHEN PAYMENTS HAVE FROZEN & OVERPAYMENT PAY SEGMENTS
Description	If the payments have both frozen and overpayment pay segments, the payment amount displayed corresponding to the overpayment and frozen payment in the Payments zone of the Remittance Summary screen is incorrect. The system displays the total payment amount instead of displaying the overpayment and frozen pay segment amount.
Workaround	None

Issue	ERROR OCCURS WHEN BIND VARIABLES USED IN IN AND NOT IN CLAUSES WITHOUT BRACKETS
Description	If a template is used in a construct for selecting usage accounts, billable charges, or adjustments where bind variables are used in the IN and NOT IN clauses without brackets, an error occurs when you bill an account through such construct.
Workaround	We recommend you to use bind variables within brackets in the IN and NOT IN clauses while defining a template.

Issue	PARTIAL DATA UPLOADED WHEN USAGE RECORD HAS MORE THAN FIVE PASS THROUGH CHARGES
Description	If you upload a usage record which has more than five pass through charges (for example, Bill Line 1, Bill Line 2, ..., Bill Line 8), at present, the system uploads the details of only five pass through charges (i.e. till Bill Line 5). The details of Bill Line 6, Bill Line 7, ..., and Bill Line 8 are not uploaded.
Workaround	None

Issue	ADJUSTMENT CREATED WHEN CONTRACT ID IS VALID, BUT ACCOUNT IDENTIFIER IS INVALID
Description	If you upload an adjustment data file with a record where the contract ID is valid, but the account ID or account identifier is invalid, the system creates the adjustment against the contract. Ideally, the system should not create the adjustment until and unless the contract ID, account ID, account identifier type, and account identifier are valid.
Workaround	None

Issue	ABLE TO GENERATE A BILL FOR A SETTLEMENT ACCOUNT FROM THE BILL SCREEN
Description	Ideally, the system should not allow you to generate a bill for a settlement account. However, at present, you can generate a bill for a settlement account from the Bill screen.
Workaround	None

Issue	ADJUSTMENT CREATED AGAINST PREVIOUS BILL IS NOT CONSIDERED DURING TRIAL BILLING
Description	If you create an adjustment against the previous completed bill, the adjustment is not presented on the next bill during trial billing. This open item accounting feature is not supported during trial billing.
Workaround	None

Issue	REOPEN BUTTON DISABLED WHEN C1-BILLSETT ALGORITHM ATTACHED ON CUSTOMER CLASS
Description	If you attach an algorithm of the C1-BILLSETT algorithm type on the Bill Completion system event of the account's customer class, at present, you will not be able to reopen a bill.
Workaround	None

Issue	THE PAYMENT REQUEST SCREEN DOES NOT SUPPORT MULTIPLE TENDERS
Description	At present, the system does not support multiple tenders when you create a payment from the Payment Request screen.
Workaround	None

Issue	PERFORMANCE ISSUE ON CHANGING THE SEARCH BY FILTER OPTION
Description	If you change the filter option from the Search By list, the system takes long time to load the respective query zone. You may observe this issue in many screens where the multi-query zone is used.
Workaround	None

Issue	EFFECTIVE PRICING NOT VISIBLE IF C1_PER_REL HAS INVALID PERSON RELATIONSHIP TYPE
Description	If you set the Check on Feature Configuration parameter in an algorithm which is created using the C1-CUSRLALGT algorithm type to A and add an invalid person relationship type in the C1_PER_REL feature configuration, the effective pricing is not inherited properly as expected.
Workaround	None

Issue	ERROR LOG FILES GENERATED ON EXECUTING BILLING AND C1-BLPPR BATCHES
Description	In ORMB, the error log file is generated even when the following batches are executed successfully and bills are completed: <ul style="list-style-type: none"> • BILLING • C1-BLPPR
Workaround	None

Issue	MANUAL DISTRIBUTION NOT WORKING IN THE PAYMENT SCREEN
Description	On saving, the distributed amount is reset to zero when you manually distribute the tender amount among the unpaid bills in the Payment screen.
Workaround	None

Issue	ERROR OCCURS ON EXECUTING BILLOPEN IF IN CLAUSE CONTAINS MULTIPLE BIND VARIABLES
Description	If you have used template in a construct where IN clause has multiple bind variables, an error occurs while executing the Construct Based - Pending Bill Generation (BILLOPEN) batch.
Workaround	None

Issue	ELIGIBILITY CRITERIA ROW IS SKIPPED IF LEFT HAND SIDE PARAMETER IS NOT SPECIFIED
Description	If the left hand side parameter is not specified in an eligibility criteria row while defining or editing a price item pricing, the eligibility criteria row is skipped and not saved in the system.
Workaround	None

Issue	POST-PROCESSING BILL SEGMENT NOT REGENERATED WHEN YOU CLICK THE GENERATE BUTTON
Description	An error occurs when you click the Generate button in the Bill Segment screen while regenerating the post-processing bill segment which is present on the pending bill.
Workaround	None

Issue	RATE CALCULATED INCORRECTLY WHEN EXCHANGE RATE IS CHANGED DURING PRICING PERIOD
Description	At present, the rate is calculated and persisted using the exchange rate which is effective on the price item pricing effective start date. Therefore, the persisted rate shown on the Pricing (Account) screen might be incorrect when the exchange rate is different on the date when the results are fetched on the Pricing (Account) screen.
Workaround	None

Issue	ENTRY NOT CREATED IN THE CI_REPRC_ENTITY_DTL TABLE ON EDITING A PRICE LIST
Description	If you edit the details of a price list which is assigned to an account or a person, an entry is not created in the CI_REPRC_ENTITY_DTL table. In other words, the repricing is not triggered on editing the details of a price list which is already assigned to an account or a person.
Workaround	None

Issue	UNABLE TO DEFINE PARAMETER FOR FIELDS WHICH BELONG TO CHILD TABLES
Description	At present, the system does not list the fields of child tables in the Source Type Code field when you select the source entity as Account, Person, or Product while defining a parameter. Therefore, you cannot define a parameter for child table' fields and as a result, repricing is not triggered when you change the value of any child table' field. For example, when you change the main customer of an account, repricing is not triggered because the MAIN_CUST_SW field belongs to the child table named CI_ACCT_PER table for which you cannot define a parameter in the system.
Workaround	None

Issue	RATE NOT PERSISTED ON PRICE ASSIGNMENT DATE WHEN C1-PRICEACCOUNT INVOKED MANUALLY
Description	When you manually execute the C1-PriceAccount business service for an account, rate is calculated and persisted for the price item pricing available on the default and global price list on the date when the C1-PriceAccount business service is invoked. Ideally, the rate must be calculated and persisted on the price item pricing effective start date.
Workaround	None

Issue	PERSISTED DATA NOT GETTING REFRESHED ON EDITING PRICE ASSIGNMENT
Description	On editing a price item pricing, the rate is not properly recalculated and persisted when the details of a price component, such as rate, eligibility criteria, and so on are changed.
Workaround	None

Issue	PRICELIST EDIT VALIDATIONS NOT DEPENDENT ON ASSIGNMENT DATE OR ASSIGNMENT STATUS
Description	Available and Eligible dates can be edited in Price List though Assignment Date or Assignment Status is added for price list.
Workaround	None

Issue	STACKING IS NOT SUPPORTED FOR PRICE SIMULATION
Description	Stacking is supported for bill generated through an Account. Pricing Simulation does not support stacking.
Workaround	None

Issue	ADJUSTMENT AMOUNT IS NOT PICKED UP FOR PAYMENT IN NEXT BILL
Description	When you generate an Adjustment after a successful payment, the Adjustment amount is not picked up in the next bill that is generated and hence, payment is not created for the adjustment amount.
Workaround	None.

Issue	SPLIT AUTO PAY NOT AVAILABLE ON SETTLEMENT CONSTRUCT ACCOUNT CREATION SCREEN.
Description	At present split auto pay is not supported on Settlement construct because Percentage field has to be added in Auto Pay Instructions section while creating a new account.
Workaround	None

Issue	SQL ERROR ON PRICE ASSIGNMENT SCREEN WITH ADHOC VALUE PASSED IN QUOTES.
Description	At present SQL error is displayed in Price Assignment screen for parameter when adhoc value is added in quotes. Price Assignment should be successful though value added is in single quotes.
Workaround	None

Issue	BS ARE NOT GETTING PICKED FOR BATCH DATE AFTER BILLABLE CHARGE END DATE
Description	At present, batch business date is later then billable charge date and C1_BILLGEN is not creating any BS, hence bill is not generated.
Workaround	None

Issue	RSDETAILS UPDATED INCORRECTLY WHEN PRICE COMPONENT FEES/RATE) PERSISTENCE EDITED
Description	At present, rate schedule details after modification are displayed and not the details before editing.
Workaround	None

Issue	FEES RATE CALCULATION PRE-PROCESSING ALGORITHM ISSUE
Description	Algorithm only considers the latest Service Quantity Identifier, it has to consider both Parameter and Service Quantity Identifier when calculating FEES and RATE.
Workaround	None

Issue	C1-ACFEES BATCH RUN WITH PRICE ITEM CODE AS THE ONLY PARAMETER.
Description	C1-ACFEES batch should consider records of price assignment where FEES for same should persist.
Workaround	None

Issue	PRICE ASSIGNMENT TYPE - POST PROCESSING ISSUE IN FEES CALCULATION BATCH.
Description	New value to be added in Price Assignment Type for post processing issue in FEES calculation batch.
Workaround	None

Issue	C1-ACCOUNTFEES SERVICE ISSUE
Description	At present FEES for price assignments are not persisted if any one price assignment eligibility rule results false and if no RATE found for same.
Workaround	None

Issue	PAYMENT STATUS REMAINS INCOMPLETE AND NO PAYMENT SEGMENT IS GENERATED EVEN WHEN TENDER CONTROL ID IS GENERATED
Description	For `On Extract Date`, if one of the split auto payments goes in to error state, BALAPY batch generates Tender Control Id and Payment Status remains Incomplete and no Payment segment is generated.
Workaround	None

Issue	CHANGES MADE TO ACCOUNT ARE NOT UPDATED WHEN APPROVAL WORKFLOW IS ACTIVE FOR ACC
Description	At present columns are not properly aligned and changes are not updated when Approval Workflow is Active. Changes done to account should be updated when approval workflow is active.
Workaround	None

Issue	INBOUND WEB SERVICE DOES NOT TRIM LEADING & TRAILING SPACES FROM INPUT PARAMETER
Description	If you pass input parameters with leading and trailing spaces to an inbound web service, the inbound web service does not trim the leading and trailing spaces from the input parameters.
Workaround	None

Issue	IF USER ADDED DEFER AUTO PAY DATE ON ACCOUNT THEN AUTO PAY WILL NOT WORK
Description	If you manually add the date in the Defer Auto Pay Date field, the automatic payment functionality will not work for the account.
Workaround	None

Issue	TOTAL OVERDUE AMOUNT IN ACCOUNT & PERSON DETAILS SECTIONS SHOW INCORRECT VALUE
Description	If you manually include a bill in more than one active overdue process through the user interface, the amount shown in the Total Overdue Amount field is incorrect when you search for an overdue process using the Person or Account details in the Delinquency Central screen.
Workaround	None

Issue	SEASONAL ADDRESS ID DOES NOT APPEAR IN THE BILL ROUTINGS TAB OF THE BILL SCREEN
Description	At present, the seasonal address which is effective at the time of billing is considered for bill routing. You can view the seasonal address details in the Bill Routings tab of the Bill screen. However, the seasonal address ID does not appear corresponding to the Address ID field.
Workaround	None

Issue	ACCOUNT OVERRIDE ADDRESS IS NOT DELETED WHEN THE BILL ROUTING RECORD IS DELETED
Description	Once you delete a bill routing record for a person from the Account screen where the Address Source field is set to Account Override , the corresponding account override address is not deleted from the system. You can still view the account override address on the screen.
Workaround	None

Issue	ERROR OCCURS ON USING A VALUE WITH HYPHEN FOR A CHARACTERISTIC TYPE
Description	At present, an error occurs on a screen where a characteristic value with hyphen (-) is defined for a characteristic type.
Workaround	None

Issue	AN ERROR OCCURS ON THE TEMPLATE SEARCH WINDOW
Description	If you do not specify at least one account selection template while defining a construct, an error occurs indicating that at least one account selection template must be specified. Now, when you search for an account selection template using the Search icon corresponding to the respective field, the Template Search window appears with the same error (indicating that at least one account selection template must be specified). The system should not display any error in the Template Search window.
Workaround	None

Issue	DISPUTE AMT AT BILL LEVEL INCORRECT WHEN BILL SEGMENT OF PREVIOUS BILL CANCELED
Description	If a bill segment of the previous bill is canceled and you create a dispute request against an account for the corresponding next bill, the dispute amount displayed against the bill in the Dispute Details zone is incorrect.
Workaround	None

Issue	SAVE BUTTON IS ENABLED WHEN A BILL IS IN THE COMPLETE OR CANCELED STATUS
Description	The Save button in the Page Title area on the Bill screen should be disabled when a bill is in the Complete or Canceled status. However, at present, the Save button is enabled when a bill is in the Complete or Canceled status.
Workaround	None

Issue	APAYCRET BATCH DOES NOT CONSIDER ECR ADJUSTMENTS CREATED ON PENDING BILL
Description	You can only use an adjustment type where the Print by Default and Impact Next Bill Balance check boxes are not selected to create transfer adjustment while distributing earnings credit rate. Therefore, an ECR adjustment created against a pending bill is not stamped on the bill. As a result, the Automatic Payment Creation (APAYCRET) batch creates automatic payment for the bill without considering the ECR adjustment.
Workaround	None

Issue	SINGLE MATCH EVENT CREATED WHEN MULTIPLE BILL SEGMENTS OF A CONTRACT ARE NETTED
Description	At present, the system creates single match event for all bill segments of a contract which are netted. For example, there are two bill segments – BS1 (50\$) and BS2 (-50\$) of the C1 contract on a bill. In this case, the system nets the BS1 and BS2 because it results in zero contract balance, and then creates single match event for BS1 and BS2.
Workaround	None

Issue	RECOGNITION SCHEDULE NOT GENERATED FOR BX/AX WHEN DRR FOR BS/AD DOES NOT EXIST
Description	There might be situations when you have attached the C1-REVRECSCH algorithm to a contract's contract type for which bill segments and adjustments are already generated. Now, if already generated bill segments or adjustments are canceled, the system will generate the deferred revenue recognition for BX and AX even if the deferred revenue recognition does not exist for the corresponding BS and AD. But, the recognition schedule is not generated. You cannot even edit the recognition schedule of a deferred revenue recognition which is created for BX and AX.
Workaround	None

Issue	AUTO PAY ID IS NOT UPDATED WHEN A REOPENED BILL IS COMPLETED
Description	When you reopen and complete a bill, the bill's due date is recalculated. On completing a reopened bill, the system does not check whether there are rule based auto pay instructions for the account which are effective on the latest bill due date. In other words, the system does not update the auto pay ID against the financial transactions in the C1_FT_EXT table. In addition, the entries in the CI_BILL_ACH table are not updated.
Workaround	None

Issue	CUSTOMER SIMULATION NOT WORKING FOR CUSTOMER HAVING LARGE DATA
Description	If a customer has large number of accounts in its hierarchy or if there are large number of billable charges for distinct price items, the corresponding prospect hierarchy will not be created successfully when you create a deal for the customer using the simulation type as Customer .
Workaround	Create a deal for such customers using the simulation type as Deal

Issue	INCORRECT AVG PRICE AND COST CALCULATION WHEN VOLUME/COMMITMENT HAVE MULTIPLE SQIS
Description	The system calculates the average price and cost incorrectly when there are multiple SQIs in the SQI-based billable charges.
Workaround	None

Issue	HIERARCHY UI-APPROVED PRICE ITEMS GETTING UNAPPROVED AGAIN IF RM CHANGES THE PRICING AND DOES THE SIMULATION AGAIN
Description	If an approver request the submitter to resubmit the deal for approval and if the submitter makes any changes in the pricing for a price item, the system should only change the status of the price item to Pending for Approval while simulating the deal. But, the system changes the status of all price items in the deal to Pending for Approval .
Workaround	None

Issue	DEAL END DATE IS NOT CONSIDERED FOR PRICE ASSIGNMENT PRICELIST ASSIGNMENT AND PRODUCT ENROLLMENT
Description	In the Apply Back feature, the system does not use the deal end date while creating price assignments, price list assignments, and product enrollments.
Workaround	None

Issue	SAME ORASEARCH ZONE APPEARS FOR MULTIPLE BIND VARIABLES WHILE DEFINING CONSTRUCT
Description	If a template has multiple bind variables and zone is specified for two or more bind variables, the system displays the same OraSearch window for all bind variables when you define a criteria in a construct. It shows the OraSearch zone of the bind variable which is added first in the template.
Workaround	None

Issue	INCONSISTENT DATA LENGTH FOR SRCH_CHAR_VAL AND ADHOC_CHAR_VAL
Description	At present, the ADHOC_CHAR_VAL and SRCH_CHAR_VAL columns have different column length. The SRCH_CHAR_VAL column can only store 50 characters. Therefore, an adhoc characteristic value above 50 characters is truncated and then stored in the SRCH_CHAR_VAL column. As a result, erroneous results appear when you search for an entity using a string from an adhoc characteristic value which is beyond 50 characters.
Workaround	None

Issue	ORASEARCH ICON FOR FK REF CHAR TYPE IS DISABLED IN AWB SCREENS
Description	At present, the Search icon in the Characteristic Value column is disabled when you select a foreign key value characteristic type in the screens which are designed using the Application Workbench (AWB).
Workaround	None

Issue	Original and Proposed Revenue Not Calculated Correctly
Description	At present, the original and proposed revenue of a price item are not calculated properly due to some rounding issue in the rate schedule API. Therefore, the average price of each price item, revenue of each account and customer, revenue from each product, division, and deal, and revenue variation calculated in a deal are not accurate.
Workaround	None

Issue	ON APPLY BACK THROWING DUPLICATE KEY ERROR
Description	If a deal of an entity is already in the Fully Orchestrated status and you create another deal for the same entity, the system throws duplicate key error when you apply back the subsequent deal.
Workaround	None

Issue	INCORRECT PROPOSED TRANSACTION VOLUME GETTING EXPORT IN PRINT DEAL PDF FORMAT
Description	If the default commitments are defined in a deal type, the system fetches default commitments for a price item and parameter combination even if the proposed commitments are available for the price item and parameter combination while extracting the details of the respective deals in the PDF format.
Workaround	None

Issue	ON DEAL CREATION IF WE REFERUSAGE FROM ANOTHER DEAL WHICH HAVE BILLABLE CHARGE ON ACCOUNT THEN COPING INCORRECT COMMITMENT I.E. TWICE OF ORIGINAL VOLUME
Description	While referring usage from another deal, the proposed commitments for a price item and parameter combination are copied incorrectly (i.e. twice the original volume).
Workaround	None

Issue	UNABLE TO READ OUT THE CUSTOMER OR ACCOUNT ID IN NUMBERS
Description	While using the Speech to Text facility in the Chatbot window, we need to read out the account or customer ID in words. For example, the customer ID 1337049295 should be read out as One Billion Three Hundred Thirty-Seven Million Forty-Nine Thousand Two Hundred Ninety-Five and not as One Three Three Seven Zero Four Nine Two Nine Five. Otherwise, erroneous results might occur.
Workaround	None

Insurance

Issue	OLD INSURANCE FEATURES ARE NOT TESTED AND VERIFIED IN ORMB VERSION 6.1.0.0.0
Description	In this release, the new policy data model is introduced. The old policy data model which is accessible to the INADMIN user group is no longer operational. The old insurance features, such as Insurance Control Central, Deferred Revenue Recognition, Account Current, Pay Plan, Group Billing, and List Bill Reconciliation are not tested and verified with the new policy data model.
Workaround	None

Issue	PERFORMANCE ISSUE WHILE CREATING DEFERRED REVENUE RECOGNITION SCHEDULE
Description	If there are large number of bill segments and adjustments for which deferred revenue recognition schedule must be generated, the system takes long time to generate deferred revenue recognition schedules.
Workaround	None

Issue	VALIDATION FOR FIDUCIARY CONTRACT MISSING DURING RECONCILIATION
Description	While changing the status of the reconciliation object to Ready To Pay, the system does not validate whether fiduciary contract exists for the group account.
Workaround	You need to ensure that group customer has fiduciary contract associated with the account through which payments can be made for the list bills.

Issue	TWO BILL SEGMENTS GENERATED WHEN REASON CODE EFFECTIVE DATE IS SAME AS BILL SEGMENT START DATE
Description	When you select the Update option from the Reason Code list and specify the reason code effective date same as the bill segment start date, the system creates two bill segments – one with prorated billed amount and another with prorated reported amount. Ideally, the system should only create one bill segment with prorated reported amount.
Workaround	None

Issue	UNABLE TO DISTRIBUTE THE REMAINING PAYMENT MANUALLY
Description	If you have distributed partial payment automatically through the Payment by Transaction screen, the system does not allow you to distribute the remaining payment manually.
Workaround	None

Issue	FOREIGN KEY AND FILE LOCATION CHARACTERISTIC TYPE NOT SUPPORTED IN UI MAPS
Description	The screens created using the UI map do not support the Foreign Key Value and File Location Value characteristic types. These screens only support the Adhoc Value and Predefined Value characteristic types.
Workaround	None

Issue	CHARACTERISTIC TYPES ARE NOT FILTERED BASED ON THE REASON CODE
Description	While editing the reconciliation object line, the characteristic types are not filtered based on the reason code that you have selected. Currently, it lists all characteristic types where the characteristic entity is set to Reason Code .
Workaround	None

Issue	RECONCILIATION DOESN'T WORK PROPERLY FOR PASS THROUGH BILLABLE CHARGES
Description	In the sample case workflow, the system checks whether the difference between the reported and billed amounts is within the tolerance limit. If so, the system must change the status of the reconciliation object line to WD-Match . However, at present, in case of pass through billable charges, the system changes the status of the reconciliation object line to Manual instead of WD-Match .
Workaround	None

Issue	ERROR OCCURS WHEN YOU RESOLVE A RECONCILIATION OBJECT LINE FROM THE CASE SCREEN
Description	The system allows you to manually resolve a reconciliation object line from the Case screen and change the status of the reconciliation object line to Manual Match . At present, an error occurs when you click the Manual Match button in the Case screen.
Workaround	None

Issue	OVERRIDE DESCRIPTION APPEARS INSTEAD OF DESCRIPTION IN THE SOURCE SYSTEM LIST
Description	At present, the override description of the source system appears in the Source System list instead of the description when you select the Policy option from the Search By list in the Customer 360-Degree View screen.
Workaround	None

Issue	UNABLE TO SELECT AUDIT EVENT TYPE IN PRT WHEN UPDATE ALL IS CONFIGURED IN AET
Description	At present, you cannot use an audit event type of the C1-Membership and C1_PERSON_BO business objects in the Age Based and Tier Based pricing rule types when the Update All option is selected in the audit event type.
Workaround	Therefore, we recommend you to select an audit event type of the C1-Membership and C1_PERSON_BO business objects where the Update All option is not selected.

Documentation

Issue	"ERROR 500--INTERNAL SERVER ERROR" - ONLINE HELP IS NOT WORKING
Description	<p>At present, an error occurs when you access online help for the following screens:</p> <ul style="list-style-type: none"> • Account Current • COBOL Program • Collection Control Central • Contract Type - Charge Type Mapping • Contract Type - Pay Plan Template Mapping • External Statement • FK Validation Summary • Pay Plan Template • Policy (P&C) • Reason Code • Reconciliation Object • Reconciliation Object Line Status • Unit of Measure
Workaround	None

Issue	ONLINE HELP NOT AVAILABLE FOR SOME SCREENS OR TABS
Description	At present, the online help is not available for the following screens: <ul style="list-style-type: none">• Collection Type• Request In addition, the online help is not available for the following tabs: <ul style="list-style-type: none">• Rate Schedule – SQ Rule Tab
Workaround	None

Technical Support

For any technical support, consult with Oracle Support, Oracle Partner, or Oracle Consulting that may be supporting your implementation or upgrade process.