

Guidewire BillingCenter®

BillingCenter New and Changed Guide

RELEASE 8.0.4

Copyright © 2001-2015 Guidewire Software, Inc. All rights reserved.

Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire Reinsurance Management, Guidewire ContactManager, Guidewire Vendor Data Management, Guidewire Client Data Management, Guidewire Rating Management, Guidewire InsuranceSuite, Guidewire ContactCenter, Guidewire Studio, Guidewire Product Designer, Guidewire Live, Guidewire DataHub, Guidewire InfoCenter, Guidewire Standard Reporting, Guidewire ExampleCenter, Guidewire Account Manager Portal, Guidewire Claim Portal, Guidewire Policyholder Portal, Gosu, Deliver Insurance Your Way, and the Guidewire logo are trademarks, service marks, or registered trademarks of Guidewire Software, Inc. in the United States and/or other countries.

All other trademarks are the property of their respective owners.

This material is confidential and proprietary to Guidewire and subject to the confidentiality terms in the applicable license agreement and/or separate nondisclosure agreement.

Guidewire products are protected by one or more United States patents.

Product Name: Guidewire BillingCenter

Product Release: 8.0.4

Document Name: BillingCenter New and Changed Guide

Document Revision: 02-July-2015

Contents

About BillingCenter Documentation	15
Conventions in This Document	16
Support	16

Part I

What's New and Changed in 8.0 Maintenance Releases

1 New and Changed in BillingCenter 8.0.4	19
New in BillingCenter 8.0.4	19
Support for Extension Properties of Type Array	19
Changed in BillingCenter 8.0.4	20
Invoice Assembler Plugin Is Called During Charge Invoicing Process	20
Changes to the BillingCenter API	20
2 New and Changed in BillingCenter 8.0.3	21
New in BillingCenter 8.0.3	21
New config.xml Parameter AllocateInvoiceNumberOnInit	21
Trouble Ticket Holds Placed on a Policy	22
Changed in BillingCenter 8.0.3	22
Support for Oracle Partitioned Indexes	22
3 New and Changed in BillingCenter 8.0.2	23
New in BillingCenter 8.0.2	23
Multicurrency Integration between BillingCenter and PolicyCenter	23
New Plugin Interface InboundIntegrationMessageReply	24
New BillingAPI Methods	24
Changed in BillingCenter 8.0.2	24
Removed Configuration Parameters	24
Integration Changes	24
4 New and Changed in BillingCenter 8.0.1	27
New in BillingCenter 8.0.1	27
Payment Allocation Plans	27
New Support for OSGi Plugins	28
Gosu Using Clause Now Supports Finally Clause	28
Changed in BillingCenter 8.0.1	28
Changes to Payment Schedule Modification	28
Changes to Account Billing Levels	29
Changes to Return Premium Plans	29
Changes to the Premium Report Billing Instruction	29
Changes to the Premium Report Customization Plugin	30
Conversion of Web Services from RPCE to WS-I	30
Removal of Agency Money Receipts from the Search Tab	31
Changes to Web Service Transaction IDs	31
Change to Inbound Integration Configuration	32
Gosu Suppress Warnings Annotation	32
Gosu Compound Assignment Operators for Logical AND and Logical OR	32

Part II**What's New and Changed in 8.0.0**

5 New and Changed in BillingCenter 8.0	35
New in BillingCenter 8.0	35
Multicurrency	35
Data Change Menu Link	36
Changing the Screen Layout	36
Changed in BillingCenter 8.0	36
Changes to Direct Billing	36
Changes to Agency Billing	37
Cash and Return Premium Handling	37
Policy-Level Billing with Cash Management	38
Context Field for Reversed Charges	38
Use Suspense Items for Payments that Do Not Match the Expected Amount	38
6 New and Changed in Configuration in 8.0	39
Important Changes to the Configuration Module	39
Changes to the Data Model	40
Changes to the Generate Dictionary Command	40
Changes to Guidewire Studio	40
Studio Now Built on IntelliJ IDEA Community Edition	40
Gosu Tester	40
Display Keys Editor	40
Studio Features That Are No Longer Available	40
Changes to Configuration Parameters	41
Removed Configuration Parameters	41
Added Configuration Parameters	41
Changed Configuration Parameters	41
Changes to Web Services	42
Changes Related to PCF Files	42
Template Page PCF Files Removed	42
Miscellaneous PCF Changes	42
Changes Related to Configuring BillingCenter Charges	44
Changes Related to Integrating with ContactManager	44
Removal of Preload.txt Lines that Include Run Level	45
7 New and Changed in Gosu in 8.0	47
New in Gosu in BillingCenter 8.0	47
New Assert Statement	47
In For Loop Declarations, Local Variable is Now Optional	48
Final Variable Initialization Separate from Declaration	48
The new Operator Now Is Optionally a Statement	49
Support Annotations on Function Parameters	49
Named Arguments in Annotations	49
New Support for Code Coverage Tools for Gosu Code	50
The Entity Touch API is Changed and No Longer Deprecated	50

Changed in Gosu in BillingCenter 8.0	50
Gosu is Now Case Sensitive In Most Cases	50
Bundle Changes from Gosu and Java	51
Gosu Map Enhancement Method Changes	51
Entity Literal Syntax is Deprecated	51
Entity Methods loadByKey, loadByPublicID, and remove Deprecated	52
Packages Changed to Reflect Public and Internal Status	52
Gosu Concurrency API Changes	53
The <> Inequality Operator Is Now Invalid	54
The Gosu Command Line Tool Built-in Editor Removed	54
Change in Gosu Named Parameters Usage	54
8 New and Changed in Integration in 8.0.....	55
New in Integration in BillingCenter 8.0	55
New Phone Number Normalizer Plugin	55
New MTOM Support for Results of Published WS-I Web Service	55
Multi-threaded Inbound Integration	56
New Messaging Destination Option for Messages Without Primary Object	56
Regenerating WSDL and XSD in Web Service Collections (regen-from-wsc)	57
New Charge Initializer Plugin	57
Changed in Integration in BillingCenter 8.0	57
RPCE Web Services Deprecated in 8.0	57
Changes to Existing Web Services	58
Important Changes for Java Code	62
Plugin Registry Changes in Studio	68
Changes to Document Management Integration	68
Contact Messaging Flow Changed	69
New PCPolicyPublicID Property in Web Services Methods	69
Changes to the BillingCenter Charge Invoicing Process	70
Changes to the Commission Plugin	70
Changes to the Payment Plan Plugin	71
9 New and Changed in System Administration in 8.0	73
New in System Administration in 8.0	73
Phone Number Normalizer Work Queue	73
Changed in System Administration in 8.0	73
Changes to Database Configuration	74
Changes to Work Queue Configuration	76
Changes to Scheduled Purge Batch Processing	76
Changes to Database Statistics Updating	76

10 New and Changed in Globalization in 8.0	79
New to Globalization in BillingCenter 8.0	79
Separation of Language and Regional Format	80
Base Configuration Language Support	80
Base Configuration Region Support	80
Base Configuration Currency Support	81
Base Configuration Geographical Support	81
Phone Number Data and Phone Validator Gosu Class	82
Automatic Address Completion and Fill-in Plugin Functionality	83
Ability to Provide a Sort Order for the Typecodes in a Typelist	83
Ability to Localize Script Parameter Descriptions	83
Language Module Installation Utility	84
Globalization-related Configuration Parameters	84
Globalization-related Typelists	84
Ability to Set First Day of Week by Region	85
New in the BillingCenter 8.0 Data Model	85
Changes to Globalization in BillingCenter 8.0	87
Changes to Address Configuration 8.0	87
Changes to Admin Data Localization	87
Changes to 8.0 Typelists	88
Changes to Files Used with Globalization	88
List Sort Methods Support Optional Locale Sensitive Sorting	90

Part III

What's New and Changed in 7.0.0

11 New and Changed in BillingCenter 7.0	93
Account Funds Tracking	93
Policy-level Billing	94
List Bill Accounts	94
Payment Instruments	94
New Producer Write-Offs Screen	94
Agency Bill Suspense Items Screen	94
Agency Bill Exception Persistence	95
Support for Enhanced Integration with PolicyCenter	95
Support for Integration with ContactManager	95
12 New and Changed in Configuration in 7.0	97
Changes to the Data Model	97
Modifications to the Address Entity	97
Modifications to Search Functionality	98
Modifications to Associative Arrays	98
Changes to Guidewire Studio	98
Modifications to Type System Refresh	99
Modifications to the Resources Pane	99
Modifications to Tools Options	99
Modifications to Dialogs	99
Changes to Configuration Parameters	99
Changes to Web Services	100

Changes Related to PCF Files	101
New iterator Attribute on Buttons	101
New disablePostOnEnter Attribute on Cell	101
New onEnter Attribute on WizardStep	101
New isDraftOnEnter Attribute	101
New Link widgets Menu Command	101
New Exposes Tab on ListViewPanel Widget	102
Deprecated Unintended ListView ContentCell Configuration	102
New Modal Cell and CellRef Widgets	103
New CheckedValuesMenuItem Iterator	103
Improvements in Localization	103
Typelist Localization	103
Display Key Localization	104
Command Line Export Translation File Import Tool	104
13 New and Changed in Gosu in 7.0	105
New in Gosu in 7.0	105
Gosu Support for Intervals	105
Gosu Support for Numeric Literals	105
New Null-safe Operators	106
New Named Function Arguments and Argument Defaults	106
Changed in Gosu in 7.0	106
Gosu Compiles to Java Virtual Machine Bytecode	107
Gosu Syntax Changes (Compile Time Issues)	107
Gosu Behavior Changes (Run Time Issues)	112
New XML APIs and Improved XSD Support Using ‘XmlElement’	116
Guidewire XML Modeler (GX Modeler) Upgraded to Use New XML API	117
Interceptors Removed	117
14 New and Changed in Integration in 7.0	119
New in Integration in 7.0	119
Payment Instrument API	119
Invoice Item Plugin	119
Funds Tracking Plugin	120
Changes in Integration in 7.0	120
WS-I Web Services	120
Suite Configuration File Overrides URLs to Guidewire Applications	125
Setting Locale in WS-I Requests for Guidewire Servers Only	126
Changes to Locally-Accessed RPCE Web Services	126
Guidewire InsuranceSuite Plugin Implementations are Versioned	127
SOAP Implementation Classes and WSDL Packages Include Version	127
Escalation and Exception Plugins	128
Changes to Messaging System Safe Ordering	128
Changes to Messaging System Resync	129
Contact Management in BillingCenter	129
Changes to Delinquency Processing Customization	130
Changes to Billing Instruction Customization	130
Changes to Billing Summaries	131
Changes to Making Payments	131
Changes to the Invoice Assembler Plugin	132
Changes to the Invoice and Related Plugins	132
Changes to the Policy Period and Policy System Plugins	133

15 New and Changed in System Administration in 7.0	135
New in System Administration in 7.0	135
DCE VM Strongly Recommended for Development Environments	135
Changed in System Administration in 7.0	137
Updated System Requirements	137
Database Configuration Changes	137
16 New and Changed in Rules in 7.0	139
Changes in Rules in BillingCenter 7.0	139

Part IV

Release Notes Archive

17 Guidewire BillingCenter 8.0.3 Release Notes.....	143
Overview	143
Release Information	144
Version Number	144
Installing This Release	144
Upgrade Information.....	144
Support	144
Major Issues and Changes	144
New and Changed Features.....	145
Improvements to List View Navigation (PL-26518)	145
Base PCF File Changes.....	145
Changes in This Release Provided in Upgrade Diff Report.....	145
Improvements and Resolved Issues	145
Application Improvements and Resolved Issues	146
Platform Improvements and Resolved Issues	148
Documentation Improvements and Resolved Issues.....	153
Known Issues and Limitations	154
BillingCenter Known Issues	154
Studio/Platform Known Issues	158
18 Guidewire BillingCenter 8.0.2 Release Notes.....	161
Overview.....	161
Release Information	162
Version Number	162
Installing This Release	162
Upgrade Information.....	162
Support	163
Major Issues and Changes	163
Base PCF File Changes.....	163
Changes in This Release Provided in Upgrade Diff Report.....	164
Improvements and Resolved Issues	165
BillingCenter Improvements and Resolved Issues	165
Platform Improvements and Resolved Issues	168
Known Issues and Limitations	173
BillingCenter Known Issues	173
Studio/Platform Known Issues	177
19 Guidewire BillingCenter 8.0.1 Release Notes.....	181
Overview.....	181
Release Information	182
Version Number	182
Installing This Release	182

Support	182
Issues and Major Changes	182
Upgrade Support	182
Base PCF File Changes	183
Changes in This Release Provided in Upgrade Diff Report	183
Improvements and General Issues	183
Known Issues and Limitations	200
BillingCenter Known Issues	200
Studio/Platform Issues	204
20 Guidewire BillingCenter 8.0.0 Release Notes.....	207
Overview.....	207
Release Information	208
Version Number	208
Installing This Release	208
Support	208
Issues and Major Changes	208
Upgrade Support	208
Base PCF File Changes	210
Changes in This Release Provided in Upgrade Diff Report	210
Known Issues and Limitations	210
BillingCenter Known Issues	210
Studio/Platform Issues	213
21 Guidewire BillingCenter 7.0.7 Release Notes.....	217
Overview.....	217
Release Information	218
Version Number	218
Installing This Release	218
Support	218
Major Issues and Changes	218
New Version of Guidewire Document Assistant ActiveX Control (PL-29937)	218
Upgrade Support	218
Base PCF File Changes	219
Changes in This Release Provided in Upgrade Diff Report	219
Improvements and Resolved Issues	219
BillingCenter Improvements and Resolved Issues	219
Known Issues and Limitations	220
BillingCenter Known Issues	220
Studio/Platform Known Issues	224
22 Guidewire BillingCenter 7.0.6 Release Notes.....	227
Overview.....	227
Release Information	228
Version Number	228
Installing This Release	228
Support	228
Issues and Major Changes	228
Upgrade Support	228
Base PCF File Changes	229
Changes in This Release Provided in Upgrade Diff Report	229
Improvements and General Issues	230
BillingCenter Improvements and General Issues	230
Platform Improvements and General Issues	232

Known Issues and Limitations	232
BillingCenter Known Issues	233
Studio/Platform Issues	237
23 Guidewire BillingCenter 7.0.5 Release Notes.....	239
Overview.....	239
Release Information	240
Version Number	240
Installing This Release	240
Support	240
Issues and Major Changes	240
Upgrade Support.....	240
Base PCF File Changes.....	241
Changes in This Release Provided in Upgrade Diff Report.....	241
Improvements and General Issues	241
BillingCenter Improvements and General Issues	241
Platform Improvements and General Issues	247
Known Issues and Limitations	252
BillingCenter Known Issues	252
Studio/Platform Issues	258
24 Guidewire BillingCenter 7.0.4 Release Notes.....	261
Overview.....	261
Release Information	262
Version Number	262
Installing This Release	262
Support	262
Issues and Major Changes	262
Upgrade Support.....	262
Base PCF File Changes.....	263
Changes in This Release Provided in Upgrade Diff Report.....	263
Improvements and General Issues	263
BillingCenter Improvements and General Issues	263
Platform Improvements and General Issues	274
Known Issues and Limitations	276
BillingCenter Known Issues	276
Studio/Platform Issues	281
25 Guidewire BillingCenter 7.0.3 Release Notes.....	285
Overview.....	285
Release Information	286
Version Number	286
Installing This Release	286
Support	286
Issues and Major Changes	286
Upgrade Support.....	286
Producer Statement Changes.....	287
Commission Earning Changes	287
Configurable Search	288
Disaster Trouble Ticket API.....	289
Base PCF File Changes.....	289
Changes in This Release Provided in Upgrade Diff Report.....	289
Improvements and General Issues	289
BillingCenter Improvements and General Issues	289
Platform Improvements and General Issues	292

Known Issues and Limitations	295
BillingCenter Known Issues	296
Studio/Platform Issues	299
26 Guidewire BillingCenter 7.0.2 Release Notes.....	301
Overview.....	301
Release Information	302
Version Number	302
Installing This Release	302
Support	302
Issues and Major Changes	302
Changes to the Logging API.....	302
Base PCF File Changes.....	302
Changes in This Release Provided in Upgrade Diff Report.....	303
Improvements and General Issues	303
BillingCenter Improvements and General Issues	303
Platform Improvements and General Issues	306
Known Issues and Limitations	311
BillingCenter Known Issues	312
Studio/Platform Issues	314
27 Guidewire BillingCenter 7.0.1 Release Notes.....	319
Overview.....	319
Release Information	320
Version Number	320
Support	320
Changes in This Release.....	320
Base PCF File Changes.....	320
Changes in This Release Provided in Upgrade Diff Report.....	320
BillingCenter Improvements and General Issues	320
Platform Improvements and General Issues	326
Known Issues and Limitations	331
BillingCenter Known Issues	331
Platform Known Issues	334
Localization Issues	337
28 Guidewire BillingCenter 7.0.0 Release Notes.....	339
Overview.....	339
Release Information	340
Version Number	340
Installing This Release	340
Changes in this Release	340
Base PCF File Changes.....	340
Changes in This Release Provided in Upgrade Diff Report.....	340
Known Issues and Limitations	340
BillingCenter Known Issues	341
Platform Known Issues	342
29 Guidewire BillingCenter 3.0.7 Release Notes.....	345
Overview.....	345
Release Information	346
Version Number	346
Support	346
Installing This Release	346

Changes in this Release	346
Base PCF File Changes.....	346
Improvements to Upgrade Diff Report.....	346
BillingCenter Improvements and General Issues	347
Platform Improvements and General Issues	350
Known Issues and Limitations	351
BillingCenter Known Issues	352
Platform Known Issues	356
30 Guidewire BillingCenter 3.0.6 Release Notes.....	359
Overview.....	359
Release Information	360
Version Number	360
Support	360
Installing This Release	360
InetSoft Evaluation	360
Changes in this Release	360
Base PCF File Changes.....	360
BillingCenter Improvements and General Issues	361
Platform Improvements and General Issues	362
Known Issues and Limitations	363
BillingCenter Issues	363
Platform Issues	367
31 Guidewire BillingCenter 3.0.5 Release Notes.....	373
Overview.....	373
Release Information	373
Version Number	374
Support	374
Installing This Release	374
InetSoft Evaluation	374
Issues and Major Changes.....	374
Changes in this Release	378
Base PCF File Changes.....	378
Improvements to Upgrade Diff Report.....	379
Improvements and General Issues	379
Known Issues and Limitations	383
BillingCenter Issues	383
Platform Issues	386
32 Guidewire BillingCenter 3.0.4 Release Notes.....	393
Overview.....	393
Release Information	393
Version Number	394
Support	394
Installing This Release	394
Upgrade Issues	394
InetSoft Evaluation	396
Changes in this Release	396
Base PCF File Changes.....	397
Improvements to Upgrade Diff Report.....	397
Improvements and General Issues	397
Known Issues and Limitations	402
33 Guidewire BillingCenter 3.0.3 Release Notes.....	413
Overview.....	413

Release Information	413
Version Number	414
Support	414
Installing This Release	414
Upgrade Issues	414
InetSoft Evaluation	414
Changes in this Release	414
Base PCF File Changes.....	414
General Issues and Improvements	415
Known Issues and Limitations	419
34 Guidewire BillingCenter 3.0.2 Release Notes.....	431
Overview.....	431
Release Information	431
Version Number	432
Support	432
Installing This Release	432
InetSoft Evaluation	432
Changes in this Release	432
Base PCF File Changes.....	432
General Issues and Improvements	433
Known Issues and Limitations	442
35 Guidewire BillingCenter 3.0.1 Release Notes.....	453
Overview.....	453
Release Information	453
Version Number	454
Support	454
Installing This Release	454
InetSoft Evaluation	454
Changes in this Release	454
Display Key Changes	454
Base PCF File Changes.....	454
General Issues and Improvements	455
Known Issues and Limitations	466
36 Guidewire BillingCenter 3.0.0 Release Notes.....	477
Overview.....	477
Release Information	477
Version Number	478
Support	478
Installing This Release	478
Changes in this Release	478
Configuration and GScript API Changes	478
Base PCF File Changes.....	478
Known Issues and Limitations	478

About BillingCenter Documentation

The following table lists the documents in BillingCenter documentation.

Document	Purpose
<i>InsuranceSuite Guide</i>	If you are new to Guidewire InsuranceSuite applications, read the <i>InsuranceSuite Guide</i> for information on the architecture of Guidewire InsuranceSuite and application integrations. The intended readers are everyone who works with Guidewire applications.
<i>Application Guide</i>	If you are new to BillingCenter or want to understand a feature, read the <i>Application Guide</i> . This guide describes features from a business perspective and provides links to other books as needed. The intended readers are everyone who works with BillingCenter.
<i>Upgrade Guide</i>	Describes how to upgrade BillingCenter from a previous major version. The intended readers are system administrators and implementation engineers who must merge base application changes into existing BillingCenter application extensions and integrations.
<i>New and Changed Guide</i>	Describes new features and changes from prior BillingCenter versions. Intended readers are business users and system administrators who want an overview of new features and changes to features. Consult the "Release Notes Archive" part of this document for changes in prior maintenance releases.
<i>Installation Guide</i>	Describes how to install BillingCenter. The intended readers are everyone who installs the application for development or for production.
<i>System Administration Guide</i>	Describes how to manage a BillingCenter system. The intended readers are system administrators responsible for managing security, backups, logging, importing user data, or application monitoring.
<i>Configuration Guide</i>	The primary reference for configuring initial implementation, data model extensions, and user interface (PCF) files. The intended readers are all IT staff and configuration engineers.
<i>Globalization Guide</i>	Describes how to configure BillingCenter for a global environment. Covers globalization topics such as global regions, languages, date and number formats, names, currencies, addresses, and phone numbers. The intended readers are configuration engineers who localize BillingCenter.
<i>Rules Guide</i>	Describes business rule methodology and the rule sets in BillingCenter Studio. The intended readers are business analysts who define business processes, as well as programmers who write business rules in Gosu.
<i>Contact Management Guide</i>	Describes how to configure Guidewire InsuranceSuite applications to integrate with ContactManager and how to manage client and vendor contacts in a single system of record. The intended readers are BillingCenter implementation engineers and ContactManager administrators.
<i>Best Practices Guide</i>	A reference of recommended design patterns for data model extensions, user interface, business rules, and Gosu programming. The intended readers are configuration engineers.
<i>Integration Guide</i>	Describes the integration architecture, concepts, and procedures for integrating BillingCenter with external systems and extending application behavior with custom programming code. The intended readers are system architects and the integration programmers who write web services code or plugin code in Gosu or Java.
<i>Gosu Reference Guide</i>	Describes the Gosu programming language. The intended readers are anyone who uses the Gosu language, including for rules and PCF configuration.
<i>Glossary</i>	Defines industry terminology and technical terms in Guidewire documentation. The intended readers are everyone who works with Guidewire applications.

Conventions in This Document

Text style	Meaning	Examples
<i>italic</i>	Emphasis, special terminology, or a book title.	A <i>destination</i> sends messages to an external system.
bold	Strong emphasis within standard text or table text.	You must define this property.
narrow bold	The name of a user interface element, such as a button name, a menu item name, or a tab name.	Next, click Submit .
<code>monospaced</code>	Literal text that you can type into code, computer output, class names, URLs, code examples, parameter names, string literals, and other objects that might appear in programming code. In code blocks, bold formatting highlights relevant sections to notice or to configure.	Get the field from the <code>Address</code> object.
<code>monospaced italic</code>	Parameter names or other variable placeholder text within URLs or other code snippets.	Use <code>getName(first, last)</code> . <code>http://SERVERNAME/a.html</code> .

Support

For assistance, visit the Guidewire Resource Portal – <http://guidewire.custhelp.com>

What's New and Changed in 8.0 Maintenance Releases

This part includes information about new and changed features in BillingCenter 8.0 maintenance releases. For information about new and changed features in 8.0.0, see “What's New and Changed in 8.0.0” on page 33.

New and Changed in BillingCenter 8.0.4

This topic includes:

- “New in BillingCenter 8.0.4” on page 19
- “Changed in BillingCenter 8.0.4” on page 20

See also

- For resolved issues and known issues in 8.0.4, see the release notes.

New in BillingCenter 8.0.4

- “Support for Extension Properties of Type Array” on page 19

Support for Extension Properties of Type Array

Extension properties are employed during the charge invoicing process. They are used when handling `ChargeInitializer`, `ChargeInitializer.Entry`, and `ChargeInstallmentChanger.Entry` objects and their associated `Charge` and `InvoiceItem` objects. An extension property assigns a property value to a future `Charge` or `InvoiceItem` object that does not exist yet, but will be created in the future. When the object is ultimately created, the property values set by the extension properties are carried over to the new object.

Earlier versions of BillingCenter provide extension properties for `Charge` and `InvoiceItem` properties of non-array types. The current version adds support for array properties.

See also

- “Extension Properties for Charge and InvoiceItem Objects” on page 429 in the *Configuration Guide*

Changed in BillingCenter 8.0.4

This topic includes:

- “Invoice Assembler Plugin Is Called During Charge Invoicing Process” on page 20
- “Changes to the BillingCenter API” on page 20

[Invoice Assembler Plugin Is Called During Charge Invoicing Process](#)

During the charge invoicing process for a new charge, the Invoice Assembler plugin is called near the end of the operation. In BillingCenter versions 8.0.0-8.0.3, the charge invoicing process did not call the Invoice Assembler plugin.

To identify that the plugin was called during the charge invoicing process, the plugin method’s `context` parameter is set to `TC_BILLINGINSTRUCTION`.

See also

- “The Charge Invoicing Process for New Charges” on page 423 in the *Configuration Guide*
- “Invoice Assembler Plugin” on page 177 in the *Integration Guide*

[Changes to the BillingCenter API](#)

This section describes BillingCenter API methods that have been deprecated or removed.

[Removed Methods: Account.initializeCharge and AccountUtil.initializeCharge](#)

To initialize a new charge, use the `BillingInstruction.buildCharge` method.

[Deprecated Method: Charge.createInvoiceItem](#)

When creating a new charge, use the `ChargeInitializer.addEntry` method. When modifying an existing charge, use the `ChargeInstallmentChanger.addEntry` method.

[Deprecated Method: Charge.setAmount](#)

For a new charge, set the amount when initializing the `ChargeInitializer` object. To change the charge amount after the charge has been executed, reverse or offset the charge and issue a new charge for the desired amount.

[Removed Method: Charge.setTAccountOwner](#)

The T-account owner of a charge cannot be set after the charge’s billing instruction has been executed. To change the T-account owner, reverse or cancel the original charge and issue a new charge with the desired T-account owner.

[Removed Method: ChargeCommissionWriteoff.setChargeCommission](#)

Use the `WriteOffFactory.createChargeCommissionWriteOff` method. To change a charge commission writeoff, reverse the original writeoff and issue a new writeoff for the appropriate `ChargeCommission`.

New and Changed in BillingCenter 8.0.3

This topic includes:

- “New in BillingCenter 8.0.3” on page 21
- “Changed in BillingCenter 8.0.3” on page 22

See also

- For resolved issues and known issues in 8.0.3, see the release notes.

New in BillingCenter 8.0.3

This topic includes:

- “New config.xml Parameter AllocateInvoiceNumberOnInit” on page 21
- “Trouble Ticket Holds Placed on a Policy” on page 22

New config.xml Parameter AllocateInvoiceNumberOnInit

In all versions of BillingCenter, an invoice number is generated whenever an invoice is created. When creating a large number of invoices, such as during multiple simultaneous policy-preview operations, the generation of invoice numbers can slow application execution. A performance gain can be achieved by eliminating unnecessary invoice number generation and, instead, generating a number only when an invoice is committed.

To generate invoice numbers for committed invoices only, set the `config.xml` parameter `AllocateInvoiceNumberOnInit` to `false`. The default setting is `true`, which generates an invoice number whenever an invoice is created.

Trouble Ticket Holds Placed on a Policy

Trouble tickets can place a disbursement hold or a payment distribution hold on an individual policy. In previous versions, such holds could be placed only on an account. For the policy-based hold to have an effect in the BillingCenter base configuration, the account must be using policy-level billing with cash separation. If the account uses any other billing level, the policy-based hold can still be placed, but will have no effect without custom configuration.

See also

- “Trouble Ticket Holds” on page 236 in the *Application Guide*

Changed in BillingCenter 8.0.3

This topic includes:

- “Support for Oracle Partitioned Indexes” on page 22

Support for Oracle Partitioned Indexes

BillingCenter now supports partitioning Oracle database indexes by hash partition, range partition, or local partition, which is partitioning based on the partitioning of the table for the index. You define index partitioning in the `database-config.xml` file by using the `<ora-index-partitioning>` XML element, which is a subelement of `<ora-index-ddl>`, which is itself a subelement of `<ora-table-ddl>`. This new subelement replaces the element `<ora-index-hash-partitioning>`, which is no longer valid.

See also

- “Configuring Index Partitioning for Oracle” on page 32 in the *Installation Guide*

New and Changed in BillingCenter 8.0.2

This topic includes:

- “New in BillingCenter 8.0.2” on page 23
- “Changed in BillingCenter 8.0.2” on page 24

See also

- For resolved issues and known issues in 8.0.2, see the release notes.

New in BillingCenter 8.0.2

This topic includes:

- “Multicurrency Integration between BillingCenter and PolicyCenter” on page 23
- “New Plugin Interface InboundIntegrationMessageReply” on page 24
- “New BillingAPI Methods” on page 24

Multicurrency Integration between BillingCenter and PolicyCenter

BillingCenter and PolicyCenter now support InsuranceSuite integration for single currency and multicurrency configurations.

See also

- “Multicurrency Integration Between BillingCenter and PolicyCenter” on page 417 in the *Application Guide*

New Plugin Interface InboundIntegrationMessageReply

You can now easily write a message reply plugin that listens to a JMS queue using WebLogic or WebSphere. There is a new plugin interface called `InboundIntegrationMessageReply`, which has all the main features of the inbound integration plugin but provided as a message reply plugin not a startable plugin.

See also

- For related upgrade information, see “`InboundIntegrationPlugin` Plugin Interface Changed Name” on page 24.

New BillingAPI Methods

To get commission plans from an external system, call the `BillingAPI` web service method `getProducerCodeInfo`. Pass a producer code public ID as an argument. The method returns a `ProducerCodeInfo` object that contains a property called `CommissionPlanInfos`. That property contains a list of `CommissionPlanInfo` objects associated with the that producer code.

To get `AgencyBillPlanInfo` objects for a producer, called the `BillingAPI` web service method `getProducerInfo`. A multicurrency producer can have multiple agency bill plans, but only one plan is allowed per currency. Also, agency bill plans are optional, so you might specify a currency for which a producer has `AgencyBillPlanInfo`.

See also

- “Policy Administration System Core Web Service APIs (`BillingAPI`)” on page 85 in the *Integration Guide*

Changed in BillingCenter 8.0.2

This topic includes:

- “Removed Configuration Parameters” on page 24
- “Integration Changes” on page 24

Removed Configuration Parameters

The following configuration parameters have been removed from BillingCenter 8.0.2:

- `BucketSizeForHistogramsOnAllIndexedColumns`
- `CollectHistogramsOnAllIndexedColumns`

See also

- “Configuring Database Statistics Generation” on page 40 in the *System Administration Guide*

Integration Changes

InboundIntegrationPlugin Plugin Interface Changed Name

Two related changes happened in this release for the inbound integration APIs:

- The `InboundIntegrationPlugin` plugin interface is now called `InboundIntegrationStartablePlugin`. Update any existing code, as well as items in the Plugins Registry in Studio.
- You can now easily write a message reply plugin that listens to a JMS queue using WebLogic or WebSphere. There is a new plugin interface called `InboundIntegrationMessageReply`, which has all the main features of the inbound integration plugin but provided as a message reply plugin not a startable plugin.

If you previously used the `InboundIntegrationPlugin` interface, you now have to determine whether to change each usage to one of two options. You may not get compile errors for usages of the `InboundIntegrationPlugin` interface because that type name is used internally. Do not use the type `InboundIntegrationPlugin` directly. Instead, search for all usages of `InboundIntegrationPlugin` and replace with one of the following choices.

- `InboundIntegrationMessageReply` – Loads as `MessageReply` plugin, and has all the methods on the `MessageReply` interface.
- `InboundIntegrationStartablePlugin` – Loads as a startable plugin, which is the same behavior as the `InboundIntegrationPlugin` in the previous release.

To use this new plugin interface, implement `InboundIntegrationMessageReply`. Remember to edit your messaging destination in Studio just like you would for a regular `MessageReply` plugin implementation.

New and Changed in BillingCenter

8.0.1

This topic includes:

- “New in BillingCenter 8.0.1” on page 27
- “Changed in BillingCenter 8.0.1” on page 28

See also

- For resolved issues and known issues in 8.0.1, see the release notes.

New in BillingCenter 8.0.1

This topic includes:

- “Payment Allocation Plans” on page 27
- “New Support for OSGi Plugins” on page 28
- “Gosu Using Clause Now Supports Finally Clause” on page 28

Payment Allocation Plans

Payment Allocation plans, new in this release, were added to improve BillingCenter cash handling options and configuration.

Payment allocation plans provide robust out-of-the-box rules to automatically apply money to receivables for direct bill policies. The purpose of a payment allocation plan is to expose common cash handling options in the user interface. Business users can control how BillingCenter allocates automatic payments, with no configuration code required to implement common requirements.

Payment allocation plans are intended to cover the majority of carrier cash handling requirements. Cash handling is configurable in Gosu. The base configuration cash handling logic is exposed so that, if necessary, you can configure cash handling by copying and modifying existing code.

See also

- “Payment Allocation Plan Overview” on page 123 in the *Application Guide*
- “Payment Allocation Plan Settings” on page 124 in the *Application Guide*
- “Payment Allocation Plan Configuration Overview” on page 437 in the *Configuration Guide*

New Support for OSGi Plugins

You can now optionally implement plugin interfaces in Java using the OSGi standard. The OSGi framework is a Java module system and service platform that helps cleanly isolate code modules and any necessary Java libraries. Guidewire recommends OSGi for all new Java plugin development. To simplify OSGi configuration, BillingCenter includes an application called IntelliJ IDEA with OSGi Editor separate from Guidewire Studio.

See also

- “Overview of BillingCenter Plugins” on page 136 in the *Integration Guide*
- “Overview of Java and OSGi Support” on page 451 in the *Integration Guide*
- “OSGi Plugin Deployment with IntelliJ IDEA with OSGi Editor” on page 469 in the *Integration Guide*

Gosu Using Clause Now Supports Finally Clause

The Gosu `using` clause syntax now has an additional feature for adding additional cleanup code. You can optionally add a `finally` clause that runs after the statement body, even if exceptions occur in the body of the `using` clause.

See also

- “Optional Use of a finally Clause with a using Clause” on page 113 in the *Gosu Reference Guide*

Changed in BillingCenter 8.0.1

This topic includes:

- “Changes to Payment Schedule Modification” on page 28
- “Changes to Account Billing Levels” on page 29
- “Changes to Return Premium Plans” on page 29
- “Changes to the Premium Report Billing Instruction” on page 29
- “Changes to the Premium Report Customization Plugin” on page 30
- “Conversion of Web Services from RPCE to WS-I” on page 30
- “Removal of Agency Money Receipts from the Search Tab” on page 31
- “Changes to Web Service Transaction IDs” on page 31
- “Change to Inbound Integration Configuration” on page 32
- “Gosu Suppress Warnings Annotation” on page 32
- “Gosu Compound Assignment Operators for Logical AND and Logical OR” on page 32

Changes to Payment Schedule Modification

You can now optionally refactor down payment items and items that are not fully paid when refactoring invoice items during a payment schedule modification.

The Change Payment Plan screen includes the following new item filters:

- **Not Fully Paid Items** – Selecting this new radio button includes items that are not fully paid in payment plan refactoring.
- **Include Down Payment Items** – Selecting this checkbox includes the down payment item payment plan refactoring. This checkbox is visible and available for editing only if **Planned Items** or **Not Fully Paid Items** is selected.

Changes to Account Billing Levels

The billing level of an account can be changed at any time. Also, accounts using policy-level billing, can choose to enable cash separation which applies incoming payments to a specific policy rather than to any policy owned by the account.

See also

“Billing Levels” on page 299 in the *Application Guide*

Changes to Return Premium Plans

A List Bill account can be the payer account for policies. When there is a return premium on a fully paid policy, the excess credit was automatically put into the List Bill payer account’s Unapplied fund. The common requirement for this scenario is that the excess return premium be sent to the owner account.

Return premium plans now provide the option to send excess return premium for List Bill policies to the owner account. Return premium plans now have a new setting called **List Bill Account Excess** with the supported values of **Policy Owner Unapplied** and **Policy Payer Unapplied**.

Changes to the Premium Report Billing Instruction

Billing instructions of type **PremiumReportBI** divide the charge based on the policy period’s payment plan.

In earlier versions, a **PremiumReportBI** charge received special handling and was always divided into a 100% down payment and zero installments. If you wish to maintain this behavior, define a payment plan override for the **PremiumReportBI** context. The override can be defined in either the BillingCenter user interface or in Gosu code.

Define a PremiumReportBI Override in the User Interface

Administrator permission is required to change a payment plan.

1. In the **Administration** tab, select the **Business Settings** → **Payment Plans** menu item. Select the relevant payment plan.
2. Click the **Edit** button.
3. Click the **Add Override** button. Select the **Premium Report BI** item.
4. In the **Premium Report BI** tab and the **Invoicing Treatment Details** section, enter the following field values:
 - Set the **Down Payment (%)** field to 100%.
 - Set the **Max # Installments** field to 0.
5. Click the **Update** button.

Define a PremiumReportBI Override in Gosu

The following Gosu code statement defines a payment plan override for **PremiumReportBI** billing instructions. The override mimics the special handling the instruction received in earlier versions. For details on each method, refer to the *Gosu API Reference*.

```
/* Define a payment plan override for PremiumReportBI billing instructions.  
 * Note: The relevantPaymentPlan variable is of type PaymentPlan.  
 */  
planOverride = ChargeSlicingBuilder()
```

```

.withDownPaymentPercent(100)
.withMaximumNumberOfInstallments(0)
.create()
relevantPaymentPlan.setOverridesFor(BillingInstruction.TC_PREMIUMREPORTBI.get(), planOverride)

```

Changes to the Premium Report Customization Plugin

The plugin method `overrideAmountOfCharge` is removed. To set an override for a `PremiumReportBI`, define the override in the BillingCenter user interface or in Gosu code. For details, see the section “Changes to the Premium Report Billing Instruction” on page 29.

Conversion of Web Services from RPCE to WS-I

The following web services were converted from RPCE to WS-I web services in version 8.0.1. WS-I is a newer format of web services. See “Web Services Introduction” on page 27 in the *Integration Guide*.

IMPORTANT RPCE web services are deprecated. Convert existing RPCE code to WS-I web services.

One important difference in the conversion is the handling of entity instance data. Because WS-I web services do not support entity data directly as method arguments or return values, many APIs now use a separate data transfer object (DTO) to encapsulate entity data. For example, instead of passing an `Address` entity directly, it might pass a Gosu class (the DTO) called `AddressDTO`. The DTO class contains only the properties in an `Address` entity instance that are necessary for the web service. For more information about DTOs in general, see “BillingCenter Data Transfer Objects (DTOs)” on page 84 in the *Integration Guide*.

If you extend the entity data model with properties that must exist in the web service for sending or receiving data, you must also extend the data transfer object. Add to the set of properties in the corresponding Gosu class with the DTO suffix. For example, if you add an important property to the `Address` entity, also add the property to the Gosu class `AddressDTO`.

WARNING Before editing Gosu class DTO files, carefully review the introductory comments at the top of each DTO file.

RPCE web service	Look in this WS-I web service	For this information
<code>IBillingCenterAPI</code>	<code>gw.webservice.policycenter.bc800.BillingAPI</code>	Billing instructions are now handled in the <code>BillingAPI</code> WS-I web service. See “Policy Administration System Core Web Service APIs (BillingAPI)” on page 85 in the <i>Integration Guide</i>
	<code>gw.webservice.bc.bc801.InvoiceDetailsAPI</code>	For methods related to invoice details, see the <code>InvoiceDetailsAPI</code> WS-I web service. See “Invoice Details Web Service APIs (InvoiceDetailsAPI)” on page 106 in the <i>Integration Guide</i> .
	<code>gw.webservice.bc.bc801.BCAPI</code>	For most methods unrelated to billing instructions or invoice details, see the <code>BCAPI</code> WS-I web service. See “Policy Administration System Core Web Service APIs (BillingAPI)” on page 85 in the <i>Integration Guide</i> .
<code>ICollateralAPI</code>	<code>gw.webservice.bc.bc801.BCAPI</code>	For web service methods related to collateral, see the <code>BCAPI</code> WS-I web service. See “Other Billing Web Service APIs (BCAPI)” on page 115 in the <i>Integration Guide</i>

RPCE web service	Look in this WS-I web service	For this information
IPolicyPeriodAPI	gw.webservice.bc.bc801.BCAPI	For web service methods related to changing billing methods, see the BCAPI WS-I web service. See "Other Billing Web Service APIs (BCAPI)" on page 115 in the <i>Integration Guide</i> .
IIInvoiceAPI	gw.webservice.bc.bc801.BCAPI	There was only one method in the RPCE web service IIInvoiceAPI. The method previewed an installment schedule. As of version 8.0.1, for this task use the BillingAPI WS-I web service methods <code>previewInstallmentPlanInvoices</code> and <code>previewInstallmentsPlanInvoices</code> . See "Other Billing Web Service APIs (BCAPI)" on page 115 in the <i>Integration Guide</i> . On a related note, there was a WS-I web service InvoiceAPI that contained the same methods as the RPCE web service IIInvoiceAPI. The InvoiceAPI web service is now deprecated. Use the BillingAPI methods instead.
IStagingTableEncryptionAPI	gw.webservice.bc.bc801.BCAPI or gw.wsdl.pl.TableImportAPI	There were only two methods in the RPCE web service IStagingTableEncryptionAPI. The methods duplicate the staging table encryption methods in the table import web service. In this release, you can find these methods in both the BCAPI and TableImportAPI web services. They do the same task, although the BCAPI version of the method <code>encryptDataOnStagingTables</code> does not return the result value. See "Other Billing Web Service APIs (BCAPI)" on page 115 in the <i>Integration Guide</i> .

Removal of Agency Money Receipts from the Search Tab

The Agency Money Receipts menu item was removed from the **Search** tab. Equivalent search results can be acquired by performing a **Payments** search and viewing the results in the **Agency Payments** tab.

Changes to Web Service Transaction IDs

This release changed one server behavior and one client behavior for WS-I web services:

- For WS-I web service configuration on the server, this release changed the behavior of the annotation `@WsdlCheckDuplicateExternalTransaction`. If you add this annotation to your implementation class, the SOAP header `<transaction_id>` is now required in each request rather than optional.
- If you have Gosu code that is a WS-I client, there is a new utility method you can call to easily set the transaction ID for the next call. On the API object, call `api.initializeExternalTransactionIdForNextUse(transactionIDString)`. After the call completes, the application removes the transaction ID from that API object, which prevents accidental duplicate use.

See also

- "Checking for Duplicate External Transaction IDs" on page 51 in the *Integration Guide*

- “Setting Guidewire Transaction IDs” on page 75 in the *Integration Guide*

Change to Inbound Integration Configuration

This release changed how to configure inbound multi-threaded integrations such as the built-in file and JMS integrations. In previous releases, you added configuration parameters in the Plugins registry in Studio. In this release, you set a single parameter `integrationservice` and then do the rest of the configuration in the new file `inbound-integration-config.xml`.

Also, the API details for file and JMS integrations changed. There is a new plugin interface called `InboundIntegrationHandlerPlugin`. Register an implementation of this plugin to add your custom logic to your file and JMS integration.

See also

- “Multi-threaded Inbound Integration Overview” on page 281 in the *Integration Guide*

Gosu Suppress Warnings Annotation

The Gosu language now provides limited support for the Java annotation `@SuppressWarnings`, which tells the compiler to suppress warnings.

See also

- “Built-in Annotations” on page 228 in the *Gosu Reference Guide*

Gosu Compound Assignment Operators for Logical AND and Logical OR

The Gosu language has two new compound assignment operators, which are operators that apply an operation to a variable then re-assign the variable to the result:

- The new operator `&&=` performs the logical AND operation to the previous value.
- The new operator `||=` performs the logical OR to the previous value.

Both operators work with the primitive type `boolean` or the object type `Boolean` on either side of the operator. For example, suppose you have two `boolean` variables called `needsUpdate` and `flagTest`. Consider the following code:

```
needsUpdate ||= flagTest
```

That expression has the meaning of

```
needsUpdate = (needsUpdate or flagTest)
```

Do not confuse these new operators with the other operators `&=` and `|=`, which apply bitwise AND and bitwise OR operations.

See also

- “Variable Assignment” on page 93 in the *Gosu Reference Guide*

What's New and Changed in 8.0.0

This part includes information about new and changed features in BillingCenter 8.0.0. For information about new and changed features in 8.0 maintenance releases, see “What’s New and Changed in 8.0 Maintenance Releases” on page 17.

New and Changed in BillingCenter 8.0

This topic includes:

- “New in BillingCenter 8.0” on page 35
- “Changed in BillingCenter 8.0” on page 36

New in BillingCenter 8.0

This topic includes:

- “Multicurrency” on page 35
- “Data Change Menu Link” on page 36
- “Changing the Screen Layout” on page 36

Multicurrency

This release supports using multiple currencies within a single instance of BillingCenter. Multicurrency support provides the ability to bill accounts and policies and pay commission in different currencies.

BillingCenter multicurrency mode does not allow the exchange or mixing of currencies. BillingCenter architecture prevents the mixing of currencies by requiring each currency and the entities associated with that currency to be in a separate *currency silo*. Money cannot move between silos. Before this release, if you wanted to use multiple currencies you had to have a separate database and BillingCenter instance for each currency. In BillingCenter 8.0, different currencies can be stored and used in the same instance by simulating separate databases inside a single database. You can think of each currency silo as a smaller database for one specific currency, stored inside the single database alongside other currency silos for other currencies. The entities in the currency silos have almost as little interaction as entities in separate databases.

There are a few exceptions to this separation. For example, a parent account can have child accounts with different currencies.

You can configure BillingCenter 8 to use either multicurrency mode or single currency mode by setting the currency display mode. BillingCenter is always a multicurrency system, even if only one currency is used and currency display mode is set to single. The data model and the business logic do not change whether the display is set to single currency (the default) or multicurrency mode. The multicurrency user interface elements are visible in the base configuration only if you enable multicurrency display mode. Even if multicurrency display is not enabled, the currency-related fields are populated within the data model. In single currency display mode, BillingCenter does not display the user interface to set those fields. There are Gosu constructors for financial (currency silo) entities, such as `Account` or `Policy`, that are available only in single currency mode. In multicurrency mode, financial entity constructors require a currency parameter.

See also

- “Multicurrency Overview” on page 269 in the *Application Guide*
- “Configuring BillingCenter with Multiple Currencies” on page 459 in the *Configuration Guide*
- “Currency-related Configuration Parameters” on page 119 in the *Globalization Guide*

Data Change Menu Link

In the **Administration** tab, the new **Data Change** menu link enables you to push important data changes to the production server. Guidewire recommends that you use this feature sparingly and only when mission-critical data needs to be updated on running production systems.

See also

- “Data Change API” on page 47 in the *System Administration Guide*

Changing the Screen Layout

You can now adjust several aspects of the screen layout, such as the size, ordering, and visibility of list view columns. These changes are stored as layout preferences, and persist across sessions until cleared.

See also

- “Changing the Screen Layout” on page 69 in the *Application Guide*

Changed in BillingCenter 8.0

This topic includes:

- “Changes to Direct Billing” on page 36
- “Changes to Agency Billing” on page 37
- “Cash and Return Premium Handling” on page 37
- “Policy-Level Billing with Cash Management” on page 38
- “Context Field for Reversed Charges” on page 38
- “Use Suspense Items for Payments that Do Not Match the Expected Amount” on page 38

Changes to Direct Billing

Eligible items for direct bill payments are selected without regard to the distribution limit for collateral charges. Earlier releases honored the distribution limit, which was incorrect behavior.

Changes to Agency Billing

Agency Bill processing is redesigned in this release to improve the usability of the user interface and to accommodate how different users work.

Agency Bill Payment Wizard

The **Agency Bill Payment** wizard is easier to use. User interaction is streamlined so that typical tasks require fewer steps to accomplish. The presentation of information in the wizard was reworked to make it easier to find relevant information.

Performance of the **Agency Bill Payment** wizard was improved for this release by implementing the following changes:

- Invoice items are not fetched unless required for distribution
- A location group has replaced the card view for payment screens

Agency Bill Credit Distribution Wizard

A new **Agency Bill Credit Distribution** wizard lets you manually override default distribution of credits. The wizard ensures that credits from negative endorsements are applied to relevant invoice items. You can also use the wizard to distribute unapplied funds. The **Agency Bill Credit Distribution** wizard uses the same screens as the **Agency Bill Payment** wizard and is functionally similar to **Direct Bill Credit Distribution** wizard.

Zero Dollar Payments Not Supported

In BillingCenter 7, to make a manual agency bill credit distribution, you used a \$0 payment. BillingCenter 8 does not support \$0 payments.

Cash and Return Premium Handling

Cash is money that is received by the carrier and is immediately available to pay receivables.

Credit is an amount the carrier owes the policyholder. Sources of credits include the most common, return premium, as well as other negative charges. Return premium is the most common source of credit.

In this release cash and credit handling are refined and improved, including the following changes and additions:

- Cash and credit are handled separately
- Return Premium plan controls credit handling
- Negative and positive charges are no longer processed together
 - Cash can be applied immediately to pay positive invoice items
 - Credit (from return premium or other negative endorsements) is allocated according to the active return premium plan
- An account can have multiple unapplied T-accounts, allowing cash to be handled separately for each policy (see “Policy-Level Billing with Cash Management” on page 38)
- Distribution logic is rewritten in Gosu
- **Distribution Method** field moved to **Account Summary** screen
- **Distribution** screen is removed

Return Premium Handling

Return Premium plans are new in this release.

The purpose of a return premium plan is to expose common credit handling options in the user interface.

A return premium plan provides a business user the means of controlling return premium handling, with no configuration code required to implement common requirements.

The return premium plan is intended to cover the majority of carrier credit handling requirements. Credit handling is configurable in Gosu. The base configuration credit handling logic is exposed so that, if necessary, you can configure credit handling by copying and modifying existing code.

See also

- “Credit Handling” on page 191 in the *Application Guide*
- “Configuring Credit Handling” on page 451 in the *Configuration Guide*

Policy-Level Billing with Cash Management

New for BillingCenter 8, policy-level billing optionally supports cash management where an incoming payment can be applied to a specific policy. Without cash management, a payment can be applied to any policy owned by the account.

See also

- “Billing Levels” on page 299 in the *Application Guide*

Context Field for Reversed Charges

In BillingCenter 8, reversed charges are performed by processing a `ReversalBillingInstruction`. In earlier releases, reversed charges were performed by processing a billing instruction of the same type as the original charge.

This change of billing instructions affects the text shown in the reversed charge’s `Context` field shown on the `Charges` screen for the account. The `Context` field displays the text defined in the billing instruction entity’s `displayName` field, such as `General` or `Renewal`. In the BillingCenter base configuration, the `displayName` field for the `ReversalBillingInstruction` entity is left empty. As a result, reversed charges listed on the `Charges` screen have empty `Context` field values. To display a `Context` value for reversed charges, define the `ReversalBillingInstruction` entity’s `displayName` field to the desired string and rebuild the data model.

See also

- “Special Purpose Billing Instructions” on page 164 in the *Application Guide*

Use Suspense Items for Payments that Do Not Match the Expected Amount

The recommended best practice when processing an apparent under- or overpayment is to use a suspense item. To maintain data validity, alternative processing methods that BillingCenter allowed in earlier releases are no longer allowed.

See also

- “Processing Payments that Do Not Match the Expected Amount” on page 288 in the *Application Guide*

New and Changed in Configuration in 8.0

This topic includes:

- “Important Changes to the Configuration Module” on page 39
- “Changes to the Data Model” on page 40
- “Changes to the Generate Dictionary Command” on page 40
- “Changes to Guidewire Studio” on page 40
- “Changes to Configuration Parameters” on page 41
- “Changes to Web Services” on page 42
- “Changes Related to PCF Files” on page 42
- “Changes Related to Configuring BillingCenter Charges” on page 44
- “Changes Related to Integrating with ContactManager” on page 44
- “Removal of Preload.txt Lines that Include Run Level” on page 45

Important Changes to the Configuration Module

In version 8.0, the structure and meaning of modules has changed.

In prior versions of BillingCenter, all your modified files were in the directory `BillingCenter/modules/configuration`, also known as the *configuration module*. All the unmodified base configuration files were stored in a separate read-only directory. The first time you edited a base configuration file, Studio copied the file to the configuration module — a process often referred to as “copy on edit”.

In other words, the configuration module previously contained:

- Your modified versions of base configuration files
- Your new configuration files

In version 8.0, the configuration module now contains all the following files in one merged hierarchy:

- Every base configuration file, whether modified or unmodified
- Your new configuration files

You can now edit all files directly in their location in the configuration module. If necessary, you can use Studio to revert any modified base configuration file to its original version. In Studio, right-click on the file name, then click **Revert to Base**.

Changes to the Data Model

The following attributes of data model entity definitions have been removed:

- `deletefk`
- `onDelete`

The following attribute of the `entity` data model element has been removed:

- `superTypeEntity`

Changes to the Generate Dictionary Command

The `gwbc regen-dictionary` has changed with a parameter to generate XML versions of the *BillingCenter Data Dictionary* and the *BillingCenter Security Dictionary*. The command also generates associated XSD files. Use the XML and XSD files to import the *Data Dictionary* and the *Security Dictionary* into third-party database design tools.

See also

- “Regenerating the Data Dictionary and Security Dictionary” on page 24 in the *Configuration Guide*

Changes to Guidewire Studio

Studio Now Built on IntelliJ IDEA Community Edition

Guidewire Studio has been completely redeveloped as extensions to the popular IntelliJ IDEA development environment. While many BillingCenter configuration editors will be familiar, navigating Studio will be a new experience. You can learn more about using IntelliJ IDEA at: <http://www.jetbrains.com/idea/documentation>

Gosu Tester

The Gosu Tester is now called the Gosu Scratchpad.

Display Keys Editor

The Display Keys editor now provides a view of the display keys in a text format. To modify display keys, you now make your changes directly to the text of the `display.properties` file. For more information, see “Using the Display Keys Editor” on page 135 in the *Configuration Guide*.

Studio Features That Are No Longer Available

The following Studio features are no longer available:

- Constraining the debugger to intercept calls only from a particular user.
- “Debug and break” to have the debugger automatically stop at the first line of Gosu code executed.

- Logging Studio diagnostic information.
- Localizing the Studio interface.
- Script Parameters editor (you now edit the XML file directly).

Changes to Configuration Parameters

Removed Configuration Parameters

The following configuration parameters have been removed in BillingCenter 8.0.0:

- AllowActiveXAutoInstall
- CacheActive
- DisableHashJoinForProximitySearch
- DisableIndexFastFullScanForProximitySearch
- EnableExtRendering
- ExperimentalBrowserPerf
- LegacyExternalEntityArraySupport
- RoundingMode (use DefaultRoundingMode instead). See “DefaultRoundingMode” on page 49 in the *Configuration Guide*.
- UseDocumentNameAsFileName

Added Configuration Parameters

The following configuration parameters have been added in BillingCenter 8.0.0:

- AlwaysShowPhoneWidgetRegionCode
- ClusterMemberPurgeDaysOld
- ClusterMemberRecordUpdateIntervalSecs
- ClusterStatisticsMonitorIntervalMins
- ConsistencyCheckerThreads
- DefaultApplicationLanguage
- DefaultNANPACountryCode
- DefaultPhoneCountryCode
- DefaultXmlExportEncryptionId
- DocumentAssistantJNLP
- JGroupsClusterChannel
- MaxAllowedDate
- MinAllowedDate
- PluginStartupTimeout
- WorkQueueThreadpoolMaxSize
- WorkQueueThreadpoolMinSize
- WorkQueueThreadsKeepAliveTime

Changed Configuration Parameters

The following configuration parameters have been changed in BillingCenter 8.0.0:

- AllowActiveX – Renamed to AllowDocumentAssistant, and default changed to `false`.
- ClusterProtocolStack – Default value changed. See “ClusterProtocolStack” on page 37 in the *Configuration Guide*.
- UISkin – Default value changed to Titanium.
- UseGuidewireActiveXControlToDisplayDocuments – Renamed to UseDocumentAssistantToDisplayDocuments.

Changes to Web Services

For changes related to web services, see “Changes to Existing Web Services” on page 58.

Changes Related to PCF Files

Template Page PCF Files Removed

The Configuration Upgrade Tool deletes PCF template pages. These pages have a `<TemplatePage>` root element. The upgrade also removes `<EntryPoint>` elements that reference template pages.

Typical uses of `<TemplatePage>` were for reports generated from a template for BillingCenter user interface users. The closest replacement for this context is a custom servlet. See “Servlets” on page 433 in the *Integration Guide*. Your servlet implementation class can use Gosu templates and render a Gosu template. See “Why Gosu Templates are Useful for Data Extraction” on page 441 in the *Integration Guide*. Also see “Template Overview” on page 359 in the *Gosu Reference Guide*.

For more programmatic (rather than user interface) access to Gosu templates, replace template pages by a custom web service. Your web service implementation class can use Gosu templates and render a Gosu template. See “Web Services Introduction” on page 27 in the *Integration Guide*.

Miscellaneous PCF Changes

The Configuration Upgrade Tool performs the following modifications to PCF files:

- Removes the `reflectOnBottom` attribute. This attribute was used to display the a virtual toolbar at the bottom of a page. The attribute was removed because the user interface needs to match the server configuration. No alternative configuration is available.
- Converts all `postOnChange` attributes on a value widget to a child `PostOnChange` node. For example, the upgrade converts:

```
<Input id="xxx" postOnChange="true" onChange="someMethod()" disablePostOnEnter="doEvaluation()"/>
```

to:

```
<Input id="xxx">
  <PostOnChange onChange="someMethod()" disablePostOnEnter="doEvaluation()"/>
</Input>
```
- Removes the `showNoneSelected` attribute from all `DetailView` inputs that are bound to a value. Setting `showNoneSelected=false` would suppress the **None Selected** option from drop-down lists and would default to the first option. This type of configuration was incorrect because the selection of the option was generally programmatically incorrect and was often used as a shortcut instead of specifying an explicit default. Verify all removals to ensure there is not any dependent logic. If there is, specify an explicit default in the page configuration.
- Removes the `showNoneSelected` attribute from all `<ValueCellType>` nodes. See the above note about removal of the `showNoneSelected` attribute from all `DetailView` inputs that are bound to a value.
- Removes the `numDataEntriesPerRow` and `transposed` attributes from `RowIteratorNode` elements. Transposed lists are a relatively rare configuration. If you had one in your configuration, use a traditional list view.
- Removes `<DetailViewPanel>` elements from `<ButtonCell>`, `<ButtonInput>`, and `<ToolbarButtonType>` elements. Detail views can no longer be embedded inside buttons.
- Converts `valueWidth` attributes on cell widgets to `value` attributes. As of 8.0, BillingCenter sizes cells by heuristics rather than content, so `valueWidth` is no longer necessary.
- If all cells in a row have the `useHeaderStyle="true"` property, the upgrade moves the property to the row level. A list can only have one header. See below.

- Updates rows to rename the `useHeaderStyle` property to `renderAsSmartHeader`. The property is renamed because the header functionality is more than styling. When a row is rendered as a smart header, all the row header interactive features are made available.
- Renames `<ContentCell>` elements to `<Link>`.
- Converts `<Cell>` elements within `<ColumnFooter>` to `<TextCell>` elements.
- Removes any element that is not a `<TextCell>` element from `<ColumnFooter>` elements.
- Removes `<ColumnHeader>` elements from `<CellType>` elements.
- Remove `<DetailViewPanel>` from `<ContentCell>`. The upgrade performs the following steps. After the automatic upgrade, review your `<ContentCell>` configurations to manually verify the configuration and make any changes. Content cells cannot have editable detail views embedded in them. Review all removals to ensure functionality. If editable content is needed within a row of data, the recommended configuration is a list detail panel.
 - For any `<ContentCell>` that contains a `<DetailViewPanel>`, the upgrade renames the `<ContentCell>` to `<FormatCell>`.
 - For other types of `<ContentCell>`, the upgrade renames the element to `<LinkCell>`.
 - Removes elements that are not allowed in the `<FormatCell>`, such as `<DetailViewPanel>` and `<InputColumn>`. This strips out unnecessary container elements. No content will be removed.
 - Renames inputs in the `<DetailViewPanel>` to `<TextInput>` unless they are `<ContentInput>`, `<TextInput>`, or `<NoteBodyInput>`.
 - Removes attributes that were allowed on specific input elements but not on `<TextInput>`.
- Removes the `useHeaderStyle` attribute from all cells that can be bound to a value. The header style in 8.0 is a lot more extensive. Smart header capabilities have been added, in addition to the styling. Header capabilities are at the row level as opposed to the cell. If you are interested in highlighting content, there are a few other ways to achieve that. Review the PCF reference for a full list of attributes for that particular cell variant.
- Removes the `compress` attribute from `<DetailViewPanel>`.
- Removes the `compress` attribute from `<ListViewPanel>`.
- Removes the `compressIfSingleChild` attribute from `<InputGroup>`.
- Comments out `<ProgressCell>` elements. This was an uncommon widget that Guidewire has removed. If you were using it on some page and would like to continue to do so, create a list detail panel, and use the `ProgressInput` in the detail section instead.
- Removes the `refreshOnProgressComplete` attribute from `<ListViewPanel>` and `<Row>` elements. This is part of the removal of the `<ProgressCell>` widget.
- Removes the following attributes from `<ChartPanel>`:
 - `bgColor`
 - `border`
 - `displayPlotOutline`
 - `orientation`
 - `sameSeriesColor`
 - `threeD`
 - `tooltip`
- Guidewire cleaned up the `<ChartPanel>` schema as a part of simplification and a move to a more interactive experience.
- Removes the following attributes from `<DomainAxis>`:
 - `autoRange`
 - `autoRangeIncludesZero`
 - `tickUnit`

- `upperMargin`
- Removes the `<Interval>` element.
- Removes the following attributes from `<RangeAxis>`:
 - `autoRange`
 - `autoRangeIncludesZero`
 - `tickUnit`
 - `upperMargin`
- Removes the `percentComplete` attribute from `<DataSeries>`.
- Removes the following from `<DualAxisDataSeries>`:
 - `autoRangeIncludesZero`
 - `lowerMargin`
 - `tickUnit`
 - `tooltip`
 - `upperMargin`
- Removes the following chart types from the `<ChartType>` enumerator:
 - `Waterfall`
 - `Gantt`
- Renames the following chart types in the `<ChartType>` enumerator:
 - `Dial` → `Gauge`
 - `Polar` → `Radar`
 - `Ring` → `Pie`
 - `StackedArea` → `Area` (there is no more distinction between a stacked vs non-stacked area)
 - `XYStep` → `XYLine`
 - `XYStepArea` → `XYArea`

Changes Related to Configuring BillingCenter Charges

Charge Handling has been enhanced in BillingCenter 8 to provide more robust validation and to ensure the integrity of charge data. With the changes to charge handling, BillingCenter performs more validation, reducing the likelihood of creation of bad data by custom configuration code.

See also

- “Changes to the BillingCenter Charge Invoicing Process” on page 70

Changes Related to Integrating with ContactManager

For changes in BillingCenter related to integrating with ContactManager and changes in ContactManager itself, see “New And Changed Features in ContactManager 8.0.0” on page 11 in the *Contact Management Guide*.

In particular, see the following table:

- “BillingCenter File Name and Package Changes for ContactManager Integration” on page 17 in the *Contact Management Guide*

Removal of Preload.txt Lines that Include Run Level

BillingCenter 7.0.x introduced a new syntax for the `preload.txt` file to load classes at a specific run level.

The new syntax is a prefix with run level name followed by two colon characters, such as:

```
MULTIUSER::gw.api.startup.PreloadActions#headerCompileAllGosuClasses
```

This syntax is unsupported in version 8.0. Before starting up the server, review your `preload.txt` file. If there are any rows with the run level syntax, remove the run level name and the two colon characters after it.

New and Changed in Gosu in 8.0

This topic includes:

- “New in Gosu in BillingCenter 8.0” on page 47
- “Changed in Gosu in BillingCenter 8.0” on page 50

New in Gosu in BillingCenter 8.0

This topic includes:

- “New Assert Statement” on page 47
- “In For Loop Declarations, Local Variable is Now Optional” on page 48
- “Final Variable Initialization Separate from Declaration” on page 48
- “The new Operator Now Is Optionally a Statement” on page 49
- “Support Annotations on Function Parameters” on page 49
- “Named Arguments in Annotations” on page 49
- “New Support for Code Coverage Tools for Gosu Code” on page 50
- “Support Annotations on Function Parameters” on page 49
- “The Entity Touch API is Changed and No Longer Deprecated” on page 50

New Assert Statement

To confirm expectations and enforce a programmatic contract, Gosu has an `assert` statement with the same semantics and syntax as in Java. By default, `assert` statements have no effect in BillingCenter. To enable assertions, you must add the `-ea` flag on the JVM that hosts the application or Studio.

See also

- “Assert Statements” on page 114 in the *Gosu Reference Guide*

In For Loop Declarations, Local Variable is Now Optional

In previous releases, `for` loop declarations had to declare a local variable:

```
for (x in 1..10) { print(x) }
```

In some cases, the body of the loop may not use the variable.

```
for (x in 1..10) { print("hello!") }
```

In version 8.0, if the type to iterate across is a Gosu *interval*, you can optionally omit the variable declaration including the word `in`. For example:

```
for (1..10) { print("hello!") }
```

There is no upgrade impact of this change.

Final Variable Initialization Separate from Declaration

In previous releases, Gosu supported declaring a local variable final using the syntax `final var` and setting a value on the same statement that declares the variable:

```
final var b : int = 10
```

In this release, the previous syntax continues to be supported. There is a new option to declare the variable final but not immediately initialize it with a value. You must set the value eventually in that function for all possible code paths.

For example, you can now write something like this:

```
function foo() {
    final var b : int
    b = 10
}
```

The Gosu compiler verifies all code paths have initialization exactly once. In other words, any `if` statements or other flow control structures must set the variable and only once.

The following code is correct:

```
function foo(a : boolean) {
    final var b : int
    if(a) {
        b = 0
    } else {
        b = 1
    }
}
```

However, if you remove the `else` branch it is invalid because as the `final` variable is initialized only if `a` is `true`.

```
function foo(a : boolean) {
    final var b : int // INVALID CODE, UNINITIALIZED IF "a" IS FALSE
    if (a) {
        b = 0
    }
}
```

There is no upgrade impact to this change.

Final Class Variables

As in previous releases, the `final` keyword can be used on class variables. The class can set the variable only once, and only from the declared class, not by subclasses.

In this release, the `final` keyword can now be used on a class variable declaration without immediately initializing the variable. If you do not immediately initialize the variable, all class constructors must initialize it in all possible code paths.

For example, the following syntax is now valid because all constructors initialize it once in each code path:

```
class TestABC {  
    final var _name : String  
  
    construct() {  
        _name = "hello"  
    }  
    construct(b : boolean){  
        _name = "there"  
    }  
}
```

However, the following is invalid because one constructor does not initialize the final variable:

```
class TestABC {  
    final var _name : String // INVALID CODE, ALL CONSTRUCTORS MUST INITIALIZE THIS IN ALL CODE PATHS  
  
    construct() { // does not initialize the variable  
    }  
    construct(b : boolean){  
        _name = "there"  
    }  
}
```

There is no upgrade impact to this change.

The new Operator Now Is Optionally a Statement

In previous releases, the `new` operator is an expression but never a statement. For example, you could type:

```
var s = new String()
```

However, you could not type as a complete statement:

```
new String()
```

In version 8.0, the `new` operator is now a statement, and the preceding example is now valid.

The `new` statement is useful if the constructor for an object saves a copy of the new object so the return value from `new` operator is unnecessary. The `new` statement may permit more concise code in these cases.

See also

- “New Is Optionally a Statement” on page 92 in the *Gosu Reference Guide*

Support Annotations on Function Parameters

The Gosu language now supports annotations on function parameters, including Gosu block declarations. In some cases, you must explicitly add `uses` lines to declare which annotation class to use. There currently are no compile time APIs for changing IDE behavior based on the annotations. You can access function parameter annotations using Java reflection APIs.

See also

- “Annotating a Class, Method, Type, Class Variable, or Argument” on page 227 in the *Gosu Reference Guide*.
- “Gosu Class Function Parameter Argument Annotations at Run Time” on page 231 in the *Gosu Reference Guide*.

Named Arguments in Annotations

Gosu annotations now support the named arguments calling convention:

```
@KnownBreak(:targetUser = "user", :targetBranch = "branch", :jira = "ABC-xxxxx")
```

For related information about named arguments, see “Named Arguments and Argument Defaults” on page 102 in the *Gosu Reference Guide*.

New Support for Code Coverage Tools for Gosu Code

Code coverage tools analyze the degree of testing of programming code. For Gosu code in Studio, BillingCenter now supports code coverage tools that use Java class files as input to bytecode analysis. Gosu compiles to Java class files. See “Code Coverage Support” on page 39 in the *Gosu Reference Guide*. This feature requires tools that use Java class files, not source code, as input.

The Entity Touch API is Changed and No Longer Deprecated

BillingCenter protects entity instances from concurrent access through a version property that exists on all *versionable entities*. In rare cases, it may be desirable to force the version number of an entity to increment even if there is no known change to the entity yet. To force BillingCenter to increment the entity version number, update the modified time, and call preupdate rules on the object, call the `touch` method on the entity instance. The method takes no arguments.

The `touch` method existed in earlier versions of BillingCenter. It eventually became deprecated was not removed. In BillingCenter 8.0, the `touch` method is restored and no longer deprecated. The version of `touch` in this release has slightly different behavior from previous releases that will affect very few customers. In 7.0, the `touch` method set the object property `BeanVersion` temporarily to `null`. In 8.0, the `touch` method immediately increments the `BeanVersion` property.

See also

- “Entity Instance Versioning and the Entity Touch API” on page 354 in the *Gosu Reference Guide*.

Changed in Gosu in BillingCenter 8.0

This topic includes:

- “Gosu is Now Case Sensitive In Most Cases” on page 50
- “Bundle Changes from Gosu and Java” on page 51
- “Gosu Map Enhancement Method Changes” on page 51
- “Entity Literal Syntax is Deprecated” on page 51
- “Entity Methods `loadByKey`, `loadByPublicID`, and `remove` Deprecated” on page 52
- “Packages Changed to Reflect Public and Internal Status” on page 52
- “Gosu Concurrency API Changes” on page 53
- “The `<>` Inequality Operator Is Now Invalid” on page 54
- “The Gosu Command Line Tool Built-in Editor Removed” on page 54
- “Change in Gosu Named Parameters Usage” on page 54

Gosu is Now Case Sensitive In Most Cases

The Gosu language is now case sensitive in most ways. In previous releases, case-sensitivity was encouraged. Now case-sensitivity is required for nearly all types. For example, if a type is declared as `MyClass`, you cannot type it as `myClass` or `myclass`.

WARNING For symbols representing Guidewire entity type data, Gosu continues to be case insensitive. However, it is strongly recommended to write your code as if entity types are case-sensitive.

During upgrade, fix any compilation errors that Studio flags due to this change.

Bundle Changes from Gosu and Java

A *bundle* refers to a programmatic abstraction of a database transaction and the set of database rows to update. In previous releases, BillingCenter represented a bundle in Gosu as the class `gw.transaction.Bundle`. The class still exists but is now in a different package. The new fully-qualified name is `gw.pl.persistence.core.Bundle`.

Typically, you do not declare the package directly in your code. Instead, you use APIs that return a bundle. You may get compilation errors during upgrade if you directly referenced the old package name. Additionally, the new `Bundle` class is callable from Java code that uses the BillingCenter 8.0 Java API.

See also

- For details about 8.0 Java API, see “Important Changes for Java Code” on page 62.
- For general information, see “Bundles and Database Transactions” on page 341 in the *Gosu Reference Guide*.

Gosu Map Enhancement Method Changes

Gosu collection classes rely on the collection classes from the Java language. However, Gosu adds enhancement methods on those objects. By using the enhanced Gosu collection classes, with a single line of code you can loop through collection items to perform actions, extract item information, or sort items. In this release, the enhancement methods changed slightly.

The map methods `filterKeys` and `filterValues` no longer exist.

These two methods were replaced by four different methods that are more flexible and named more clearly. The following table documents the new methods. The methods that have `retain` in the name are destructive. The methods with `filter` in the name create a new map and do not modify the original map.

Map method name	Description
<code>retainWhereKeys(keyFilter)</code>	Destructively removes all entries whose keys do not satisfy the <code>keyFilter</code> expression. Returns <code>true</code> if and only if this map changed as a result of the block. The key filter block must take one argument (a key) and return <code>true</code> or <code>false</code> .
<code>retainWhereValues(valueFilter)</code>	Destructively removes all entries whose values do not satisfy the <code>valueFilter</code> expression. Return <code>true</code> if this map changed as a result of the call. The value filter block must take one argument (a value) and return <code>true</code> or <code>false</code> .
<code>filterByKeys(keyFilter)</code>	Returns a new map that is a clone of the original map but without entries whose keys do not satisfy the <code>keyFilter</code> expression. The key filter block must take one argument (a key) and return <code>true</code> or <code>false</code> .
<code>filterByValues(valueFilter)</code>	Returns a new map that is a clone of the original map but without entries whose values do not satisfy the <code>valueFilter</code> expression. The key filter block must take one argument (a value) and return <code>true</code> or <code>false</code> .

Entity Literal Syntax is Deprecated

Entity literals have been deprecated. Use query builders instead.

Entity literals have the form:

```
entityType(entityId)
```

For example:

```
User("34" /* John Smith */)
Group("3" /* Eastern Regional Center */)
```

For example, if you choose a Group or User from Studio SmartHelp in previous releases, Studio inserted an entity literal of that type into your code.

Replace these expressions with query builder expressions.

For example:

```
Query.make(Account).compare(Account#PublicID, Relop.Equals, myPublicId).select().AtMostOneRow
```

See also

- “Query Builder APIs” on page 129 in the *Gosu Reference Guide*

Entity Methods `loadByKey`, `loadByPublicID`, and `remove` Deprecated

The `Bundle` class methods `loadByKey`, `loadByPublicID`, and `remove` are deprecated in version 8.0.

Instead of using `loadByKey`, use the new method `loadBean`, which takes a `gw.pl.persistence.core.Key` object. Create a Key object with a constructor that takes the entity type and the numeric ID. For example:

```
a = gw.transaction.Transaction.getCurrent().loadBean(new Key(Address, 123))
```

Instead of using `loadByPublicID`, use the query builder APIs. For example:

```
Query.make(Account).compare(Account#PublicID, Relop.Equals, myPublicId).select().AtMostOneRow
```

Instead of using `remove`, use the `Bundle` class method `delete`.

See also

- “Query Builder APIs” on page 129 in the *Gosu Reference Guide*
- “Getting an Entity from a Public ID or a Key (Internal ID)” on page 346 in the *Gosu Reference Guide*

Packages Changed to Reflect Public and Internal Status

Many types not otherwise mentioned changed the package in version 8.0 to better reflect the status of the public or internal nature of the type.

The general rule is:

- Types in the package `com.guidewire.*` are internal. Do not use.
- Types in the `gw.*` package but not in a subpackage that contains the word `internal` are generally for customer use.

IMPORTANT If in doubt whether a class or API is supported in BillingCenter, contact Guidewire Customer Support.

Some classes changed from `com.guidewire.*` to `gw.*`. Refer to the following table for an example of most commonly used packages that changed.

Old package	New package
<code>com.guidewire</code>	<code>gw.pl.exception</code>
<code>com.guidewire.commons.file</code>	<code>gw.pl.file</code>
<code>com.guidewire.commons.io</code>	<code>gw.pl.io</code>
<code>com.guidewire.commons.util</code>	<code>gw.pl.util</code>
<code>com.guidewire.logging</code>	<code>gw.pl.logging</code>
<code>com.guidewire.main</code>	<code>gw.pl.simpletool</code>
<code>com.guidewire.modules</code>	<code>gw.pl.modules</code>
<code>com.guidewire.modules.pom</code>	<code>gw.pl.modules.pom</code>
<code>com.guidewire.testharness</code>	<code>gw.pl.testharness</code>
<code>com.guidewire.util</code>	<code>gw.pl.util</code>
<code>com.guidewire.util.csv</code>	<code>gw.pl.util.csv</code>
<code>com.guidewire.util.diff.lcs</code>	<code>gw.pl.util.lcs</code>
<code>com.guidewire.util.webservices</code>	<code>gw.pl.util.webservices</code>

Old package	New package
com.guidewire.util.webservices.axis	gw.pl.util.webservices.axis
com.guidewire.util.webservices.login	gw.pl.util.webservices.login
com.guidewire.util.xml	gw.pl.util.xml
com.guidewire.util.xml.helpers	gw.pl.util.xml.helpers
com.guidewire.pl.plugin	gw.pl.plugin
com.guidewire.pl.plugin.addressbook	gw.pl.plugin.addressbook
com.guidewire.pl.plugin.approval	gw.pl.plugin.approval
com.guidewire.pl.plugin.credentials	gw.pl.plugin.credentials
com.guidewire.pl.plugin.custompicker	gw.pl.plugin.custompicker
com.guidewire.pl.plugin.dbauth	gw.pl.plugin.dbauth
com.guidewire.pl.plugin.document	gw.pl.plugin.document
com.guidewire.pl.plugin.management	gw.pl.plugin.management
com.guidewire.pl.plugin.messaging	gw.pl.plugin.messaging
com.guidewire.pl.plugin.search	gw.pl.plugin.search
com.guidewire.pl.plugin.security	gw.pl.plugin.security
com.guidewire.pl.plugin.util	gw.pl.plugin.util
com.guidewire.pl.plugin..validation	gw.pl.plugin.validation
com.guidewire.main	gw.pl.main
com.guidewire.external	gw.pl.external

You may get compilation errors due to these changes. In many cases, Studio can help you identify the correct package using the ALT-Enter keyboard shortcut.

For details of which packages changed in version 8.0, consult the API Differences file (`apidiffs.txt`) in the product.

Gosu Concurrency API Changes

Gosu has multiple APIs related to concurrency.

Scopes APIs Deprecated

In version 7.0, the `gw.api.web.Scopes` API created maps pre-scoped to scopes such as request and session.

In version 8.0, the `gw.api.web.Scopes` API are now deprecated. Instead, use the `RequestVar` and `SessionVar` classes in the `gw.api.web` package. The `RequestVar` and `SessionVar` classes create a variable with a well-defined lifecycle and attachment point (either the request or session). It is strongly recommended to use `RequestVar` and `SessionVar` rather than the deprecated APIs or using the Java thread local API `java.lang.ThreadLocal<TYPE>`.

LazyVar Split Into Two Variants for Locking and Non-locking

In version 7.0, the `LazyVar` class implements what some people call a *lazy variable*. This means Gosu constructs it only the first time some code uses it. Because the `LazyVar` class uses the Java concurrency libraries, access to the lazy variable is thread-safe.

In version 8.0, the `LazyVar` class is removed. You must change all usages to one of two replacements:

- `LocklessLazyVar` – A lazy variable initialization that does not require thread safety.
- `LockingLazyVar` – A more direct replacement for `LazyVar`. Access to the lazy variable is thread-safe.

See also

- “Concurrency” on page 381 in the *Gosu Reference Guide*.

The <> Inequality Operator Is Now Invalid

In 7.0, you could indicate inequality using either <> or != expressions.

In 8.0, <> is invalid. Change usages to !=.

There is a code inspection option in Studio to detect *equality issues* that finds this issue.

The Gosu Command Line Tool Built-in Editor Removed

The Gosu command line tool in the BillingCenter/admin/bin directory no longer has a built-in Gosu editor. If you run the gosu.cmd tool with no arguments, Gosu launches by default into the Gosu interactive shell.

See also

- “Gosu Command Line Shell” on page 389 in the *Gosu Reference Guide*.

Change in Gosu Named Parameters Usage

In 7.0, a function call (including all method calls) that included named parameters could use a mix of named and unnamed parameters.

In 8.0, if a function call includes a named parameter, all following parameters must be named parameters.

See “Named Arguments and Argument Defaults” on page 102 in the *Gosu Reference Guide*.

New and Changed in Integration in 8.0

This topic includes:

- “New in Integration in BillingCenter 8.0” on page 55
- “Changed in Integration in BillingCenter 8.0” on page 57

New in Integration in BillingCenter 8.0

This topic includes:

- “New Phone Number Normalizer Plugin” on page 55
- “New MTOM Support for Results of Published WS-I Web Service” on page 55
- “Multi-threaded Inbound Integration” on page 56
- “New Messaging Destination Option for Messages Without Primary Object” on page 56
- “Regenerating WSDL and XSD in Web Service Collections (regen-from-wsc)” on page 57
- “New Charge Initializer Plugin” on page 57

New Phone Number Normalizer Plugin

The new globalization framework for phone numbers includes a plugin to normalize phone numbers that users enter through the application and that enter the database through data import.

See also

- “Phone Number Normalizer Plugin” on page 266 in the *Integration Guide*

New MTOM Support for Results of Published WS-I Web Service

The W3C Message Transmission Optimization Mechanism (MTOM) is a method of efficiently sending binary data to and from web services as attachments outside the normal response body.

The following table compares version 7.0 and version 8.0 MTOM support.

Role of BillingCenter	Sending or receiving	Version 7.0 support	Version 8.0 support
BillingCenter consuming a WS-I web service	Data with initial request	--	--
	Data in response	Yes	Yes
BillingCenter publishing a WS-I web service	Data with initial request	Yes	Yes
	Data in response	--	Yes, configurable

See also

- For information about MTOM where Gosu is the SOAP client. See “MTOM Attachments with Gosu as Web Service Client” on page 81 in the *Integration Guide*.
- For information about MTOM where Gosu is the SOAP server. See “Web Service Invocation Context” on page 38 in the *Integration Guide*.

Multi-threaded Inbound Integration

There may be cases that require high-performance data throughput for inbound integrations that require special threading or transaction features from the hosting J2EE/JEE application environment. It is difficult to interact with the application server’s transactional facilities and write correct, thread-safe, high-performing code. BillingCenter includes tools that help you write such inbound integrations. You can focus on your own business logic rather than how to write thread-safe code that works safely in each application server.

The inbound integration API consists of several parts:

- A plugin interface called `InboundIntegrationPlugin` that defines the contract between BillingCenter and inbound integration high-performance multi-threaded processing of input data.
- A built-in implementation of the plugin that reads text data in local files. Poll a directory in the local file system for new files at a specified interval. Send new files to integration code and process incoming files line by line.
- A built-in implementation of the plugin that reads inbound messages from a JMS message queue.

See also

- “Multi-threaded Inbound Integration” on page 281 in the *Integration Guide*

New Messaging Destination Option for Messages Without Primary Object

In the BillingCenter messaging system, many messages are associated with a primary object such as a account. For messages without a primary object, documentation refers to those messages as non-safe-ordered messages.

In version 7.0, there was a messaging destination option called Strict Mode. Strict Mode was either enabled or disabled using a check box in the user interface.

- If Strict Mode was enabled, messages without a primary object were sent in a strict order, and waited for an acknowledgement before sending the next message.
- If Strict Mode was disabled, messages without a primary object were immediately sent in a single thread, and did not wait for an acknowledgement before proceeding to the next message.

In version 8.0, both those modes are available but there is a third choice, which is a multi-threaded sending with no ordering and no waiting for an acknowledgement. The precise order of sending of messages without a primary object is non-deterministic.

The messaging editor offers three choices for the option **Messages Without Primary**:

- Single thread
- Multi thread

- Strict Mode

See also

- “Message Ordering and Multi-Threaded Sending” on page 337 in the *Integration Guide*

Regenerating WSDL and XSD in Web Service Collections (regen-from-wsc)

Web service collection (.wsc) files encapsulate the set of resources necessary to connect to a web service on an external system. If you view a web service collection in Studio and click the **Fetch Updates** button, Studio retrieves WSDL and XSD files from the servers that publish those web services. In this release, you can trigger the fetch updates process from a command line tool called `regen-from-wsc`.

See also

- “Loading WSDL Locally Using Studio Web Service Collections” on page 66 in the *Integration Guide*

See also

New Charge Initializer Plugin

The new ChargeInitializer plugin provides consistent handling of charge creation and clarifies what is allowed when creating a new charge.

See also

- “Changes to the BillingCenter Charge Invoicing Process” on page 70
- “Changes to the Commission Plugin” on page 70
- “Changes to the Payment Plan Plugin” on page 71
- “Configuring the Charge Invoicing Process” on page 423 in the *Configuration Guide*
- “ChargeInitializer Plugin” on page 172 in the *Integration Guide*
- “Payment Plan Plugin” on page 174 in the *Integration Guide*

Changed in Integration in BillingCenter 8.0

This topic includes:

- “RPCE Web Services Deprecated in 8.0” on page 57
- “Changes to Existing Web Services” on page 58
- “Important Changes for Java Code” on page 62
- “Plugin Registry Changes in Studio” on page 68
- “Changes to Document Management Integration” on page 68
- “Contact Messaging Flow Changed” on page 69
- “New PCPolicyPublicID Property in Web Services Methods” on page 69
- “Changes to the BillingCenter Charge Invoicing Process” on page 70
- “Changes to the Commission Plugin” on page 70
- “Changes to the Payment Plan Plugin” on page 71

RPCE Web Services Deprecated in 8.0

BillingCenter has two types of web services:

- **WS-I web services** – WS-I web services use SOAP protocol and are compatible with the WS-I standard.

- **RPC Encoded (RPCE) web services** – RPCE web services are an older style of web service publishing APIs. Versions of BillingCenter before version 7.0 support only RPCE web service publishing.

In version 7.0, Guidewire recommended that you publish and consume web services using the WS-I APIs rather than the RPCE APIs. Although support for RPCE web services was for backward compatibility only, RPCE web services were not officially deprecated.

In version 8.0, RPCE web services are deprecated. RPCE web services continue to work in this release. However, it is strongly recommended to immediately start to convert RPCE code to the WS-I equivalent code.

Built-in RPCE Web Services Converted to WS-I Web Services

In conjunction with deprecation of RPCE web services, some built-in RPCE web services now have WS-I equivalents. The WS-I versions are similar in structure as their RPCE equivalents. However, in some cases the parameter types changed to accommodate differences between RPCE and WS-I. For example, WS-I does not support entity types directly as arguments or return types. In some cases, entity type arguments or return types are replaced with Gosu classes or XML objects that represent the same information.

The service name changed slightly for some web services, for example to remove the initial I from the name. The package names are different for the WS-I web services, and include the version number of BillingCenter. As with all WS-I web services, use CNTL-N to find one or more implementation classes in Studio. If more than one is available, check the package name. For example, if the package contains bc/800 or p1/800, that version was introduced in BillingCenter 8.0.0.

For a summary of changes to specific web services, including the name of any RPCE web services it replaces, see “Changes to Existing Web Services” on page 58.

Changes to Existing Web Services

For details of converted WS-I web services, refer to the documentation topic for each web service as specified in the following table. Note that all RPCE web services are deprecated in version 8.0. Begin to convert any RPCE web service code to WS-I.

For any web services that converted from RPCE to WS-I, the WS-I versions use different types so there will be code changes. For example, any arguments or return values that took entity types now are WS-I variants based on Gosu classes with those fields or XSD types.

WS-I web service	Any RPCE web services that it replaces	Description and notable changes in version 8.0
BillingAPI	n/a	<p>Some actions related to billing. There is a new method called <code>getUnapplieds</code> to retrieve all the unapplied fund amounts for the given account, as an array of <code>unappliedfund</code>. See “Policy Administration System Core Web Service APIs (BillingAPI)” on page 85 in the <i>Integration Guide</i>.</p>
ContactAPI	n/a	<p>Manipulates contact data. For customers who use ContactManager, ContactManager calls this web service.</p> <ul style="list-style-type: none"> In version 7.0, some methods took a transaction ID, which represents a unique identifier for this request. If the request is duplicated later, the application knows that it is a duplicate and does not repeat the request. In 8.0, the methods no longer have the transaction ID as an explicit method argument. Instead, the methods have the annotation <code>@WsICheckDuplicateExternalTransaction</code>, which automatically enforces unique transaction IDs based on SOAP headers. This change affects the methods <code>updateContact</code>, <code>removeContact</code>, and <code>mergeContacts</code>. See “Checking for Duplicate External Transaction IDs” on page 51 in the <i>Integration Guide</i>. If the web service client is another Guidewire application, you can set the transaction ID with the API <code>ContactAPIUtil.setTransactionId()</code>. New methods required by the ABCClientAPI, which the web service implements: <code>pendingUpdateRejected</code>, <code>pendingCreateRejected</code>, <code>pendingCreateApproved</code>, <code>pendingUpdateApproved</code>. These methods manage pending updates but the implementation is different for each Guidewire application. <p>In BillingCenter, PolicyCenter does not generate pending updates or creation for ContactManager, so this callback is not called in the default configuration of both applications. BillingCenter throws an exception.</p>
ImportToolsAPI	IIImportToolsAPI IUserAPI IGroupAPI	<p>See “Contact Web Service APIs” on page 401 in the <i>Integration Guide</i>.</p> <p>In addition to converting IIImportToolsAPI from RPCE to WS-I, the ImportToolsAPI web service has new methods:</p> <ul style="list-style-type: none"> <code>importXml</code> – For administrative data only, this method imports data from XML conforming to either an archiving import xsd, or from <code>bc_import.xsd</code>. <code>importArchiveXmlData</code> – new method to import administrative data from XML data in UTF-8 encoding of archiving data format XML. <code>importArchiveXmlDataAsByteArray</code> – new method to import administrative data from XML data in a byte array of archiving data format XML <p>From WS-I web services if you want to add a new user or group, use the WS-I web service ImportToolsAPI. Do not use the old web services specific to users and groups: IUserAPI and IGroupAPI. You can add a user or group with methods that import XML or CSV formatted text.</p> <p>For other methods in the RPCE web services IUserAPI and IGroupAPI, there is no equivalent WS-I web service. If you used IUserAPI or IGroupAPI, Guidewire recommends that you write your own custom WS-I web service that transfers only the necessary data for each integration point.</p>

WS-I web service	Any RPCE web services that it replaces	Description and notable changes in version 8.0
LoginAPI	ILoginAPI (however, see note)	<p>Although the names are similar, the RPCE web service ILoginAPI and the WS-I web service LoginAPI are actually notably different.</p> <p>The RPCE web service ILoginAPI had a critical role in authenticating with the server for later API calls with other web services. If you use the generated Java libraries and the included APILocator utility class, APILocator indirectly uses ILoginAPI.</p> <p>The WS-I web service LoginAPI does not have the analogous role in WS-I authentication. In effect, WS-I authentication happens with each API call. However, if your web service client code wants to explicitly test specific authentication credentials, BillingCenter publishes the built-in LoginAPI web service. Call this web service's login method, which takes a user name as a String and a password as a String. If authentication fails, the API throws an exception. You can also use LoginAPI to purposely leave a user session open for logging purposes.</p> <p>See “Login Authentication Confirmation” on page 58 in the <i>Integration Guide</i>.</p>
MaintenanceToolsAPI	IMaintenanceToolsAPI	<p>Tools related to managing the application. There are various changes in 8.0 in addition to conversion to WS-I:</p> <ul style="list-style-type: none"> • terminateBatchProcessByName changed name to terminateBatchProcessByName • terminateBatchProcessByID changed name to requestTerminationOfBatchProcessByID • stopWorkQueueWorkers – new method to stop query workers • getWQueueStatus – new method that replaces getWorkQueueStatus. • getNumActiveWorkItems – new method to get number of active work items. <p>See “Maintenance Tools Web Service” on page 128 in the <i>Integration Guide</i>.</p>
MessagingToolsAPI	IMessagingToolsAPI	<p>Tools related to messaging, such as suspending a destination. There are various changes in 8.0 in addition to conversion to WS-I:</p> <ul style="list-style-type: none"> • configureDestination – new method to configure some performance and error handling aspects of a messaging destination on a running server. Also see the getConfiguration method. • getConfiguration – get performance and error handling parameters from a messaging destination on a running server. Also see the configureDestination method. <p>There is also a new messaging_tools command line tool with similar functionality. Use the -config parameter to get the configuration for a destination. Use the -restart parameter to restart the messaging destination with new settings. If you use -restart, add additional options: wait, retries, initial, backoff, poll, threads, chunk.</p> <p>See “Messaging Tools Web Service” on page 360 in the <i>Integration Guide</i>.</p>

WS-I web service	Any RPCE web services that it replaces	Description and notable changes in version 8.0
PaymentAPI	IBillingCenterAPI	<p>Payment related actions. There were various changes in 8.0:</p> <ul style="list-style-type: none"> • <code>makeDirectBillPayment</code> – method no longer takes the account ID argument. • <code>makeDirectBillPaymentToPolicyPeriod</code> – method no longer takes the account ID argument. • <code>makeAccountAdjustment</code> – for the adjustment amount, the method takes a <code>MonetaryAmount</code> rather than a <code>Number</code>. • <code>payToProducerUnapplied</code> – new method to pay to producer unapplied • <code>writeoffProducerUnapplied</code> – new method to write off producer unapplied • <code>makeAgencyBillPayment</code> – new method to make an agency bill payment • <code>makeAgencyBillPaymentToPolicyPeriod</code> – new method to make an agency bill payment to a policy period • <code>makeDirectBillPaymentToInvoice</code> – new method to make a direct bill payment to an invoice
		<p>See “Payments Web Service APIs (PaymentAPI)” on page 107 in the <i>Integration Guide</i>.</p>
ProfilerAPI	IProfilerAPI	<p>There are various changes in 8.0 in addition to conversion to WS-I:</p> <ul style="list-style-type: none"> • <code>setEnableProfilerForBatchProcess</code> – the method argument of type <code>BatchProcessType</code> now is <code>String</code> • <code>setEnableProfilerForWorkQueue</code> – the method argument of type <code>BatchProcessType</code> now is <code>String</code>.
		<p>See “Profiling Web Services” on page 131 in the <i>Integration Guide</i>.</p>
SystemToolsAPI	ISystemToolsAPI	<p>There are various changes in 8.0 in addition to conversion to WS-I:</p> <ul style="list-style-type: none"> • <code>getDBCCState</code> – new method that returns the state of the consistency checks process. • <code>submitTDBCCBatchJob</code> – new method that submits the consistency checks batch job on the underlying physical database • <code>submitUpdateStats</code> – new method that runs the update stats process on the underlying physical database. Afterwards, use the <code>isUpdateStatsRunning</code> method to check on the process status. • <code>cancelUpdateStats</code> – new method that cancels the update stats process if running. Afterwards, use the <code>getUpdateStatsState</code> method to check on the process status. • <code>getUpdateStatsState</code> – new method that checks whether the update stats process is currently running. • <code>submitDmvReportBatchJobWithParams</code> - the previous version used to take three parameters, and now takes only the <code>includeDatabaseStatistics</code> argument. • <code>getRecentAWRSnapshotInfo</code> - method now returns <code>DatabaseSnapshotInfo[]</code> instead of <code>String</code> • <code>updateLoggingLevel</code> -- logging level was passed as <code>String</code>, now is <code>LoggingLevel</code> • <code>submitAwrReportBatchJob</code> and <code>submitAwrReportBatchJobAllpts</code> – new method to submit Oracle AWR Performance Report batch jobs. • <code>getPerfReportsInfo</code> – Retrieve the information about the latest Oracle AWR downloads.
		<p>See “System Tools Web Services” on page 129 in the <i>Integration Guide</i></p>

WS-I web service	Any RPCE web services that it replaces	Description and notable changes in version 8.0
TypeListToolsAPI	ITypeListToolsAPI	"Mapping Typecodes to External System Codes" on page 125 in the <i>Integration Guide</i>
WorkflowAPI	IWorkflowAPI	<p>There are various changes in 8.0 in addition to conversion to WS-I:</p> <ul style="list-style-type: none"> • invokeTrigger - method argument that used to be type <code>WorkflowTriggerKey</code> now is <code>String</code>. • isTriggerAvailable – method argument that used to be type <code>WorkflowTriggerKey</code> now is <code>String</code> <p>See "Workflow Web Services" on page 130 in the <i>Integration Guide</i>.</p>

Important Changes for Java Code

BillingCenter 8.0 made significant changes to how you write and deploy Java code, including Java implementations of plugin interfaces. The Java API from previous releases is deprecated in version 8.0.

If you deployed Java code in previous releases:

- Your legacy code that uses the deprecated Java API runs with no changes in BillingCenter 8.0.
- To regenerate the libraries for the legacy API, continue to use the same `gwbc regen-java-api` command as in previous releases. However, there you must add an additional flag on the tool to generate the deprecated Java libraries in addition to the 8.0 Java API libraries. Also, BillingCenter generates the libraries and Javadoc for the deprecated API in a slightly different location. See "Java Library Generation in Version 8.0" on page 65.
- Although the older API is deprecated, you can continue to use the deprecated API in this release while you start to upgrade your code to the new API.
- You can use Java code from the deprecated API and code using the new API at the same time for different plugin interfaces. Be sure that for each plugin interface and Java class, you follow deploy requirements for each API style (deprecated API or new API).
- The location on disk of a Java class or library determines whether BillingCenter expects the deprecated Java API or the version 8.0 Java API.
 - For the 8.0 Java API, deploy classes in the locations defined in "Deploying Non-OSGi Java Classes and JARs" on page 468 in the *Integration Guide*. Most notably, you must deploy all classes and JAR files in directories that have `/basic/` in the path. Any other locations are unsupported with the 8.0 Java API.
 - For the deprecated Java API, deploy classes and JARs in a plugin directory as defined in previous releases of BillingCenter. BillingCenter loads the class with the legacy class loader and does the external entity conversion and legacy conversion of containers (collections and lists). In general, it is incorrect to put any classes or libraries that use the deprecated Java API into any `plugins/PLUGINDIR/basic/lib` folder or `plugins/PLUGINDIR/basic/classes` subdirectories. The `PLUGINDIR` value represents the plugin directory as specified for that plugin implementation in the Studio Plugins editor. See "Adding an Implementation to a Plugins Registry Item" on page 110 in the *Configuration Guide*. The one exception is if you have classes or libraries that do not use Guidewire entity data and previously were in the `plugins/shared/basic` subdirectory.
 - For a complete reference, see "Where to Deploy Your Java Classes and Libraries" on page 66.
 - For more about plugin directories, see "Adding an Implementation to a Plugins Registry Item" on page 110 in the *Configuration Guide*.

Overview of Java API Changes in BillingCenter 8.0

There are many differences between the deprecated Java API and the BillingCenter 8.0 Java API. The most important difference is that there is no longer a proxy layer between Gosu code and your Java code. In the deprecated API, your Java code would be written against a Java API defined solely within the generated entity libraries.

In the 8.0 Java API, the set of supported types is the set of types that are all of the following:

- types written in Java
- types in the `gw.*` packages but not in the unsupported `gw.internal.*` packages

Types in any other packages are unsupported. For example, never use classes in the `com.guidewire` package, which is internal.

Some supported types have some unsupported fields and methods. The unsupported features are marked with the `@gw.lang.InternalAPI` annotation.

IMPORTANT To identify types and features with the `@InternalAPI` annotation in your external Java IDE, see “Inspections to Flag Unsupported Internal Java APIs” on page 453 in the *Integration Guide*.

The packages for BillingCenter entity types appear differently in Java than in Gosu. See “Entity Packages and Customer Extensions from Java in Version 8.0” on page 64.

Other important aspects of new Java API are different:

- There is no need for special handling of collections across the proxy boundary between Gosu and Java. In previous releases, by default BillingCenter performed a shallow clone of collections that were function arguments or return values. In more recent versions of BillingCenter, there were ways of omitting this behavior, but is no longer ever necessary. In the 8.0 Java API, you can use Java collection classes from Java as you would normally.
- There is no need to map additional Java types to access from your Java code.
- There is no mapping of Gosu package namespaces to proxied types. Types created in Gosu are available only through reflection. See “Calling Gosu Classes from Java” on page 66.
- Object properties and methods work directly on Java types. For Gosu types, object properties and methods are available only through reflection. See “Calling Gosu Classes from Java” on page 66.
- Static properties and methods work directly on Java types. For Gosu types, static properties are available only through reflection. See “Calling Gosu Classes from Java” on page 66.
- In previous releases, Gosu enhancement properties and methods appeared directly on the type. In the new API, Gosu enhancements on Java types are available only through reflection. See “Gosu Enhancements on Java Classes or Gosu Classes” on page 66.
- In previous releases, exceptions that BillingCenter throws are wrapped in a different exception type such as `com.guidewire.external.GenericCheckedException` or `java.lang.RuntimeException`. In the 8.0 Java API, exception handling in Java is straightforward. Just use `try`, `catch`, and `rethrow` as you normally would in Java.
- There is no stripping of parameterization from types in the new Java API. In other words, suppose a plugin interface uses the generics language feature to declare an argument type as `ArrayList<MyClass>`. In the legacy API, it appeared in Java simply as `ArrayList`. In the new Java API, it retains its fully-parameterized type `ArrayList<MyClass>`.

In general, in the new Java API you can use Java classes, interfaces, and other types simply by referencing them directly in Java. However, there are important changes discussed in later topics:

- “Entity Packages and Customer Extensions from Java in Version 8.0” on page 64
- “Java Library Generation in Version 8.0” on page 65
- “Getting a Reference to a Bundle in Java” on page 65

- “Creating Entities in the BillingCenter 8.0 Java API” on page 65
- “Calling Gosu Classes from Java” on page 66
- “Gosu Enhancements on Java Classes or Gosu Classes” on page 66

Do Not Create Java Directly in Studio

It is unsupported to add Java class files in the Studio user interface. However, due to technical reasons, BillingCenter Studio cannot hide user interface options that can add Java classes to the file hierarchies.

IMPORTANT Do not create Java classes directly in Studio. It is unsupported. If you want to code in Java, you must use a separate IDE for Java development. For example, use a separate instance of IntelliJ IDEA or Eclipse just for your Java development.

Avoiding Internal APIs

The Java API allows you to use the same Java types that you can use in Gosu. However, Guidewire specifies some methods and fields on these types for internal use only, and unsupported for customer use. In Gosu, these *internal API* methods and fields are hidden, and use of them triggers a compilation error. In Java, when you are using your own IDE separate from Studio, internal APIs are visible even though unsupported.

Guidewire indicates internal API methods and properties with the annotation `@gw.lang.InternalAPI`.

If you use IntelliJ IDEA as your Java IDE, configure the IDE to flag inappropriate uses of internal APIs as warnings. Your copy of BillingCenter Studio includes a code inspection you can use. See “Inspections to Flag Unsupported Internal Java APIs” on page 453 in the *Integration Guide*.

Entity Packages and Customer Extensions from Java in Version 8.0

Accessing entity and typelist data from Java changed significantly in this release. For complete documentation on entity and typecode data from Java, see “Accessing Entity and Typecode Data in Java” on page 455 in the *Integration Guide*.

In Gosu, you can refer to an entity type using the syntax simply `entity.ENTITYNAME` or simply the entity name because the package `entity` is always in scope. In the BillingCenter Java API, you can reference a type directly by its fully-qualified name. However, for BillingCenter entity types, from Java the fully-qualified name of an entity is not simply `entity.ENTITYNAME` or simply the entity name. The syntax `entity.ENTITYNAME` or using the entity name with no package is a shortcut within the Gosu language’s type system.

Unlike in previous releases, BillingCenter exposes each entity type as up to three different interfaces. Which interface name to use depends on whether the property is defined in the base configuration, application-specific extensions, customer extensions, or entirely new entities that you define.

If you only want the base configuration properties, the type name is the same in Java as in Gosu, but the package varies by entity type. Some aspects of the fully-qualified names of some interfaces are configurable.

For typecodes, there are differences for how to access a typecode based on where the typecode was defined or extended. This is similar to the differences mentioned earlier for multiple entity interfaces in the 8.0 Java API.

For more information, refer to the following sections:

- “Accessing Entity and Typecode Data in Java” on page 455 in the *Integration Guide*
- “Entity Packages and Customer Extensions from Java” on page 456 in the *Integration Guide*
- “Typecode Classes from Java” on page 460 in the *Integration Guide*.

WARNING The Java entity interfaces are supported only in Java code. From Gosu, use the syntax `entity.ENTITYNAME`, or where unambiguous you can simply type `ENTITYNAME`.

Java Library Generation in Version 8.0

As in the legacy Java API, in the new Java API there is a library generation step. After you make changes to the data model to add additional extension properties, with both APIs you must regenerate the libraries to use extension properties.

BillingCenter 8.0 Java API

For the 8.0 Java API, run the `regen-java-api` command from the `BillingCenter/bin` directory:

```
gwbc regen-java-api
```

BillingCenter generates the libraries and Javadoc for the 8.0 Java API at the location:

```
BillingCenter/java-api/lib  
BillingCenter/java-api/doc
```

See “Regenerating Java API Libraries” on page 455 in the *Integration Guide*.

Deprecated Java API

To use the deprecated Java API, you must add an additional parameter to the `regen-java-api` command:

```
gwbc regen-java-api -Ddeprecated=true
```

BillingCenter generates separate deprecated libraries and Javadoc in addition to the regular libraries and Javadoc. The output is in a slightly different location than in previous releases:

```
BillingCenter/java-api/deprecated/lib  
BillingCenter/java-api/deprecated/doc
```

Note that Java code that uses the deprecated Java API must use the plugin directory locations defined in previous versions of BillingCenter with the deprecated API. For a complete reference, see “Where to Deploy Your Java Classes and Libraries” on page 66. For more about plugin directories, see “Adding an Implementation to a Plugins Registry Item” on page 110 in the *Configuration Guide*.

Getting a Reference to a Bundle in Java

To use entity instances, in many cases you need a reference to a *bundle*. A bundle is a programmatic abstraction that represents one database transaction. See “Bundles and Database Transactions” on page 341 in the *Gosu Reference Guide*. The package for the `Bundle` class changed in this release for both Gosu and Java.

To get the current bundle using the BillingCenter 8.0 Java API, use the following code:

```
gw.p1.persistence.core.Bundle b = gw.transaction.Transaction.getCurrent();
```

If there is no current bundle, you must create a bundle before creating entity instances or updating entity instances that you get from a database query. See “Getting a Reference to an Existing Bundle in Java” on page 464 in the *Integration Guide*.

Creating Entities in the BillingCenter 8.0 Java API

In previous releases, you would create new entity instances in Java using the `EntityFactory` class. The `EntityFactory` class is now deprecated. Only use the `EntityFactory` with the deprecated Java API.

In the BillingCenter 8.0 Java API, there are two ways to create a new entity instance.

The recommended API for creating an entity instance is to call the `newInstance` method on the entity type’s `TYPE` property. Pass a bundle reference as a method argument. For example:

```
// if you do not need customer extension properties....  
Address a1 = Address.TYPE.newInstance(b);  
  
// if you need customer extension properties, use this syntax...  
AddressExt a2 = (AddressExt) Address.TYPE.newInstance(b);
```

For a longer code example and alternative APIs for this task, see “Creating New Entity Instances from Java” on page 464 in the *Integration Guide*.

Calling Gosu Classes from Java

In the 8.0 Java API, you can call Gosu classes from Java. However, the syntax and API is very different. You must use *reflection* (introspection of the type system at run time) instead of direct access to types and properties at compile time. See “Accessing Gosu Classes from Java Using Reflection” on page 466 in the *Integration Guide*.

Gosu Enhancements on Java Classes or Gosu Classes

You can call Gosu enhancements defined on Java classes or on Gosu classes. However, the syntax and API is very different. You must use *reflection* (introspection of the type system at run time) instead of direct access to types and properties at compile time. See “Gosu Enhancement Properties and Methods in Java” on page 467 in the *Integration Guide*.

Exception Handling Now Straightforward

With the legacy Java API, exceptions that BillingCenter throws would be wrapped in a different exception type such as `com.guidewire.external.GenericCheckedException` or `java.lang.RuntimeException`.

In the 8.0 Java API, exception handling in Java is straightforward. Just use `try`, `catch`, and `rethrow` as you normally would in Java.

Typecode Static Properties Have the TC_ Prefix, Just Like from Gosu

In the 8.0 Java API, the static properties on a typelist that represent a typecode have the `TC_` prefix, just like from Gosu. However, to actually work with the typecode, you must call the `get` method on the static property to get the appropriate typecode object. For example:

```
LossType.TC_OPEN.get()
```

For additional information about typecodes in the new API, see “Typecode Classes from Java” on page 460 in the *Integration Guide*.

EntityMock is Deprecated

The `EntityMock` API is part of the legacy Java API and is now deprecated. There is no direct replacement for it, but you can write your own mock code to test against your Java code.

Querying for Entities By Public ID

In the legacy Java API, there was an API to find entity instances by public ID:

```
myClaim = EntityFactory.getInstance().getEntityByRef(Claim.class, "cc:1234")
```

In the 8.0 Java API, just use the standard query builder APIs. See “Query Builder APIs” on page 129 in the *Gosu Reference Guide*.

For minor differences in the Gosu and Java APIs for querying, see “Querying for Entity Data in Java” on page 466 in the *Integration Guide*.

Where to Deploy Your Java Classes and Libraries

Implementing Java Plugins

If you are writing Java classes that implement BillingCenter plugin interfaces, those are known as *Java plugins*.

In the following directory listings, `PLUGINDIR_OR_SHARED` represents that the value can be either:

- a plugin directory name as defined in the plugin registry for that plugin
- the special value `shared`. Use `shared` if other Java class loaders or plugin directories need access to those classes or libraries.

For more about plugin directories, see “Adding an Implementation to a Plugins Registry Item” on page 110 in the *Configuration Guide*.

If the code from version 7.0 uses entity data for anything, use the following locations:

- Put classes in `BillingCenter/modules/configuration/plugins/PLUGINDIR_OR_SHARED/classes`
- Put libraries in `BillingCenter/modules/configuration/plugins/PLUGINDIR_OR_SHARED/lib`

From the previous two locations, BillingCenter always performs legacy external entity conversion and container (list or collection) conversion.

Note: If your Java plugin code from version 7.0 uses Guidewire entity data, as in 7.0 you must copy the generated file `gw-entity-cc.jar` to the directory `BillingCenter/modules/configuration/plugins/shared/lib`.

If the code from version 7.0 does not use entity data for anything, use the following locations:

- Put classes in `BillingCenter/modules/configuration/plugins/shared/basic/classes`
- Put libraries in `BillingCenter/modules/configuration/plugins/shared/basic/lib`

From the previous two locations, BillingCenter never performs legacy external entity conversion or container (list or collection) conversion. You cannot use these locations if you use any entity data with the deprecated API.

If the code uses the BillingCenter 8.0 Java API, use the following locations:

- Put classes in `BillingCenter/modules/configuration/plugins/PLUGINDIR_OR_SHARED/basic/classes`
- Put libraries in `BillingCenter/modules/configuration/plugins/PLUGINDIR_OR_SHARED/basic/lib`

In the Java API 8.0, external entity conversion and container conversion is unnecessary.

Java Classes and Libraries to Call from Arbitrary Gosu Code

In version 8.0, as in previous releases, you can deploy Java classes and libraries for use by arbitrary Gosu code in Studio. In other words, in addition to writing plugin code in Java, you can write other Java classes and call them from Gosu.

In the following directory listings, `GOSU_OR_SHARED` represents that the value can be either Gosu or shared. Carefully note the capitalization of the two words. Use shared if other Java class loaders or plugin directories need access to those classes or libraries.

If the code from version 7.0 uses entity data for anything, use the following locations:

- Put classes in `BillingCenter/modules/configuration/plugins/GOSU_OR_SHARED/classes`
- Put libraries in `BillingCenter/modules/configuration/plugins/GOSU_OR_SHARED/lib`

From the previous two locations, BillingCenter always performs legacy external entity conversion and container (list or collection) conversion.

If the code from version 7.0 does not use entity data for anything, use the following locations:

- Put classes in `BillingCenter/modules/configuration/plugins/Gosu/classes`
- Put libraries in `BillingCenter/modules/configuration/plugins/Gosu/lib`

From the previous two locations, BillingCenter always performs legacy external entity conversion and container (list or collection) conversion.

- Put classes in `BillingCenter/modules/configuration/plugins/shared/basic/classes`
- Put libraries in `BillingCenter/modules/configuration/plugins/shared/basic/lib`

From the previous two locations, BillingCenter never performs legacy external entity conversion or container (list or collection) conversion. You cannot use these locations if you use any entity data with the deprecated API.

If the code uses the BillingCenter 8.0 Java API, use the following locations:

- Put classes in `BillingCenter/modules/configuration/plugins/GOSU_OR_SHARED/basic/classes`

- Put libraries in `BillingCenter/modules/configuration/plugins/GOSU_OR_SHARED/basic/lib`

In the Java API 8.0, external entity conversion and container conversion is unnecessary.

Plugin Registry Changes in Studio

Due to larger changes in BillingCenter Studio, the user interface changed for registering plugins. See “Using the Plugins Registry Editor” on page 109 in the *Configuration Guide*.

In version 8.0, there are several options for registering code that implements a plugin interface, but the deployment options vary:

- **Gosu** – a Gosu class
- **Java** – a Java class. BillingCenter uses the location of classes and libraries on disk to determine whether to use the legacy Java class loader or the BillingCenter 8.0 Java class loader. For a complete reference, see “Where to Deploy Your Java Classes and Libraries” on page 66. For the 8.0 Java API, also see “Deploying Non-OSGi Java Classes and JARs” on page 468 in the *Integration Guide*.

Changes to Document Management Integration

Document Assistant is a Java Applet

In previous releases, the Document Assistant was an ActiveX plugin.

In 8.0, the Document Assistant is a Java applet. As such, it no longer strictly requires Microsoft Internet Explorer. The user web client must have Java installed. For the list of supported browsers, Java versions, and operating systems, see the *Guidewire Platform Support Matrix*, available from the Guidewire Resource Portal at <https://guidewire.custhelp.com/app/resources/products/platform>.

It is no longer necessary to make PCF edits to the file `NewTemplateDocumentDV.pcf` when enabling or disabling the Document Assistant.

See also

- “Configuring Guidewire Document Assistant” on page 147 in the *System Administration Guide*
- “Document Production” on page 231 in the *Integration Guide*

Client-side PDF Document Production Removed

In previous releases, you could optionally do PDF document production either client-side or server-side. In conjunction with other changes to document production, in version 8.0 the client-side PDF document production is unsupported.

Server-side PDF document production, continues to work unchanged in this release.

Client-side production on Windows for Microsoft Word and Microsoft Excel work unchanged in this release. However, the client-side production is managed by the new Java applet instead of an ActiveX control. In version 8.0, the Document Assistant is a Java applet that encapsulates client-side scripts. On Windows, the client-side scripts are JScript merge scripts for Windows versions of Microsoft Word and Microsoft Excel.

Document Content Response Types Removed: HTML and JSCRIPT

In the previous release there were four document content response types. Each content response type is a value in the `ResponseType` property within `DocumentContentsInfo`.

In version 7.0, the response types were:

- Raw document contents as an input stream. Response type is `DOCUMENT_CONTENTS`
- A web page containing an ActiveX control. Response type is `HTML_PAGE`

- JScript code to run on the user's machine. Response type is **JSCRIPT**.
- A URL that can display the content from a local content store. Response type is **URL**.

In version 8.0, HTML and JScript document production types were removed.

In version 8.0, the only response types are:

- Raw document contents as an input stream. Response type is **DOCUMENT_CONTENTS**
- A URL that can display the content from a local content store. Response type is **URL**.

Removal of JScript Response Type Does Not Affect Typical JScript Use

Note that although the JScript response type was removed, this change does not affect typical client-side document production using JScript on Windows that used Document Assistant. In version 8.0, the Document Assistant is now a Java applet and encapsulates the client-side scripts such as JScript merge scripts for Windows applications. Similar to previous releases, you can use the **URL** response type to display the Java applet.

Configuration Parameter Changes

Because of changes to the Document Assistant, the following `config.xml` parameters changed:

- The parameter `AllowActiveX` was removed and replaced with a new parameter `AllowDocumentAssistant`. The value of the old parameter name is ignored.
- The parameter `UseGuidewireActiveXControlToDisplayDocuments` changed names to `UseDocumentAssistantToDisplayDocuments`. The value of the old parameter is used for the new parameter.
- The parameter `AllowActiveXAutoInstall` was removed.
- The parameter `UseDocumentNameAsFileName` was removed.

Minor Change in Document Management Mailmerge Attribute

There is a document template descriptor called `mailmergetype`. It optionally configures of pagination of client-side Microsoft Word production. By default, BillingCenter uses Microsoft Word *catalog pagination*, which correctly trims the extra blank page at the end. However, catalog pagination forbids template substitution in headers and footers. In contrast, *standard pagination* adds a blank page to the end of the file but enables template substitution in headers and footers.

In 7.0, you set the attribute to `catalog` for catalog pagination and any other value to use standard pagination.

In 8.0, set this attribute to the value `catalog` to use catalog pagination. To use standard pagination, do not set this attribute at all.

If you set this value to some value other than `catalog`, change your document template descriptor to not set this attribute.

Contact Messaging Flow Changed

The messaging flow changed for contact-related messages to external systems. There are different methods on the `ContactSystemPlugin` interface.

See also

- For more information about the new messaging flow for contacts, see “Integrating with a Contact Management System” on page 395 in the *Integration Guide*.

New PCPolicyPublicID Property in Web Services Methods

BillingCenter billing instructions include a new important property `PCPolicyPublicID`. This property contains the public ID of the policy in PolicyCenter (or other policy system). This property is used in the integration between PolicyCenter and BillingCenter.

Similarly, PolicyCenter web services have the new property. BillingCenter plugins that send information to a policy system now also have the `PCPolicyPublicID` property.

Changes to the BillingCenter Charge Invoicing Process

The New ChargeInitializer Class and Charge Initializer Plugin

The handling of charges has been enhanced in BillingCenter 8 to provide enhanced validation and data integrity.

New charges are handled by the new `ChargeInitializer` class and can be customized with the new `ChargeInitializer` plugin.

Existing charges are customized by using the new `ChargeInstallmentChanger` class.

See also

- “Configuring the Charge Invoicing Process” on page 423 in the *Configuration Guide*
- “ChargeInitializer Plugin” on page 172 in the *Integration Guide*

The `InvoiceItem.setAmount` Method Is No Longer Callable

`InvoiceItem` objects are created at the conclusion of the charge-handling process when the charge is executed. Until this late stage, `InvoiceItem` objects do not exist. Prior to the creation of `InvoiceItem` objects, `Entry` objects are created and act as fledgling invoice items. Customization is performed on the `Entry` objects. When the charge is finally executed, the `Entry` objects are converted to `InvoiceItem` objects. The `Entry` objects associated with a charge are accessible from the `ChargeInitializer` and `ChargeInstallmentChanger` objects.

Because invoice item properties are not assigned directly to an `InvoiceItem` object, the ability to assign a monetary amount to an `InvoiceItem` object is no longer supported. Accordingly, the `InvoiceItem.setAmount` method can no longer be called. Existing code that directly sets the monetary amount of an `InvoiceItem` will no longer compile and must be modified to use the new `Entry` object.

Uncallable Method	Replacement Method
<code>InvoiceItem.setAmount</code>	<code>Entry.setAmount</code> Entry objects are contained within the <code>ChargeInitializer</code> and <code>ChargeInstallmentChanger</code> objects.

Changes to the Commission Plugin

The following sections describe deprecated Commission plugin methods.

Note: The deprecated methods are no longer invoked by BillingCenter. Existing customization code in the deprecated methods is effectively orphaned and must be relocated to their respective replacement methods.

Deprecated Method: `getNetAmountForPaymentReceivedPayableCalculator`

The `getNetAmountForPaymentReceivedPayableCalculator` method is replaced by the new `shouldWriteoffActiveCommissionForChargeWrittenOff` method.

Deprecated Method	Replacement Method
<code>getNetAmountForPaymentReceivedPayableCalculator</code>	<code>shouldWriteoffActiveCommissionForChargeWrittenOff</code>

The replacement method is a placeholder for customization code. In the base configuration, the method always returns `false`.

Deprecated Method: `shouldMakeCommissionPayable`

The `shouldMakeCommissionPayable` method was provided with two signatures. Both method signatures are deprecated. No replacement methods are provided.

Deprecated Methods	Replacement Methods
<code>shouldMakeCommissionPayable(itemCommission : ItemCommission, amountToMakePayable : MonetaryAmount) : Boolean</code>	None
<code>shouldMakeCommissionPayable(policyCommission : PolicyCommission, amountToMakePayable : MonetaryAmount) : Boolean</code>	None

Changes to the Payment Plan Plugin

The following sections describe deprecated and modified Payment Plan plugin methods.

Note: The deprecated methods are no longer invoked by BillingCenter. Existing customization code in the deprecated methods is effectively orphaned and must be relocated to their respective replacement methods.

Method Signature Change: `createFullSetOfInstallmentEventDates`

The `createFullSetOfInstallmentEventDates` method has changed the datatype of its first parameter. The original parameter was of type `Charge`. The new parameter is of type `ChargeInitializer`.

Original Method Signature	New Method Signature
<code>createFullSetOfInstallmentEventDates(charge : Charge, ...)</code>	<code>createFullSetOfInstallmentEventDates(initializer: ChargeInitializer, ...)</code>

Deprecated Method: `createFullSetOfInvoiceItems`

The replacement method for the deprecated `createFullSetOfInvoiceItems` method depends on the scenario in which the method is called.

Deprecated Method Scenario	Replacement Method
To process new charges	<code>ChargeInitializer.customizeChargeInitializer</code> in the <code>ChargeInitializer</code> plugin
To process invoice items that were recalculated due to a payment plan change	<code>PaymentPlan.recreateInvoiceItemsForPaymentPlanChange</code> in the <code>Payment Plan</code> plugin

Deprecated Method: `recreatePlannedInvoiceItemsForPaymentPlanChange`

The `recreatePlannedInvoiceItemsForPaymentPlanChange` method is replaced by the new `recreateInvoiceItemsForPaymentPlanChange` method. The new method accepts a `ChargeInstallmentChanger` object.

Deprecated Method	Replacement Method
<code>recreatePlannedInvoiceItemsForPaymentPlanChange</code>	<code>recreateInvoiceItemsForPaymentPlanChange</code>

The list of invoice items passed to and returned by the deprecated method is stored as a list of `Entry` objects in the `entries` property of the `ChargeInstallmentChanger` object. The `Entry` objects are converted to invoice items when the charge is executed.

The deprecated method recreated only *planned* invoice items. Also, the method was invoked only when processing *planned* invoice items for a changed payment plan. The replacement method receives a group of recreated invoice items that may include all or a subset of existing invoice items. The group of items is not restricted to planned items only. Also, the replacement method is invoked whenever a payment plan is changed and not only when processing planned items.

When moving customization code from the deprecated to the replacement method, any assumptions in the code about handling only planned invoice items must be examined for appropriate behavior. The customization code in the replacement method must be able to handle any existing invoice item, not only planned items.

Changed Method: `recreatePlannedInstallmentEventDatesForPaymentPlanChange`

The `recreatePlannedInstallmentEventDatesForPaymentPlanChange` method recreates a group of installment item event dates. The group of event dates is received in the method's parameter list.

In BillingCenter 8.0, the group of dates can include *any* event date, not just planned dates (despite the method's name). In earlier releases, the method recreated only planned event dates.

Also in BillingCenter 8.0, the method can be invoked whenever the payment plan is changed. In earlier releases, the method was invoked only when a changed payment plan recreated *planned* charges.

Existing code may have to be modified if it contains assumptions about handling only planned items or about being invoked only when processing planned items.

See also:

- “Payment Plan Plugin” on page 174 in the *Integration Guide*

New and Changed in System Administration in 8.0

This topic includes:

- “New in System Administration in 8.0” on page 73
- “Changed in System Administration in 8.0” on page 73

New in System Administration in 8.0

This topic includes:

- “Phone Number Normalizer Work Queue” on page 73

Phone Number Normalizer Work Queue

BillingCenter 8.0 includes a Phone Number Normalizer work queue. This work queue calls the registered plugin that implements the `IPhoneNormalizerPlugin` interface to convert phone numbers from the BillingCenter 7.0 single-field standard to the BillingCenter 8.0 multi-field standard.

See also

- “Upgrading Phone Numbers” on page 185 in the *Upgrade Guide*

See also

Changed in System Administration in 8.0

This topic includes:

- “Changes to Database Configuration” on page 74

- “Changes to Work Queue Configuration” on page 76
- “Changes to Scheduled Purge Batch Processing” on page 76
- “Changes to Database Statistics Updating” on page 76

Changes to Database Configuration

As of BillingCenter 8.0, the `<database>` element is located in a separate file, `database-config.xml`, rather than the `config.xml` file. The configuration upgrade includes an automated step to modify the database configuration. Guidewire has made the following changes to the database configuration:

- The `<database>` element, now in `database-config.xml`, no longer contains subelements with the following syntax:

```
<param name="name" value="value">
```
- For Oracle, the `<tablespacemapping>` elements have been replaced with a single `<tablespaces>` element. The `<tablespaces>` element is contained in an `<ora-db-ddl>` parent element. The `<tablespaces>` element includes the attributes `admin`, `index`, `op`, `staging`, `typelist`, and `lob`. These attributes correspond to the logical tablespaces defined in BillingCenter. You can use these attributes to map tablespaces that you have created to the logical tablespaces.
- For SQL Server, the `<tablespacemapping>` elements have been replaced with a single `<mssql-filegroups>` element. The `<mssql-filegroups>` element is contained in an `<mssql-db-ddl>` parent element. The `<mssql-filegroups>` element includes the attributes `admin`, `index`, `op`, `staging`, `typelist`, and `lob`. These attributes correspond to the logical tablespaces defined in BillingCenter. You can use these attributes to map file groups that you have created to the logical tablespaces.
- If a `<tablegroup>` element was contained in a `<database>` element that had an `env` attribute defined, the upgrade copies the `env` attribute onto the `<tablegroup>` element.
- If any of the following `<database>` attributes are defined, the upgrade copies them over to the `<database>` element in `database-config.xml`: `addforeignkeys`, `autoupgrade`, `checker`, `dbtype`, `env`, `name`, `printcommands`. The schema for these attributes has not changed.
- If any comments exist within the `<database>` element, the upgrade copies these comments to the `<database>` element in `database-config.xml`.
- If the `driver` attribute of the `<database>` element equals `dbcp`, the upgrade adds a `<dbcp-connection-pool>` element and copies the `jdbcUrl` parameter to the `jdbc-url` attribute of the `<dbcp-connection-pool>` element. If the original configuration did not include a `jdbcUrl` parameter, then the upgrade logs an error. If a `passwordFile` attribute is specified on the `<database>` element of the old configuration, the upgrade transfers the `passwordFile` attribute to the `<dbcp-connection-pool>` element. The upgrade converts any of the following parameters defined in the old configuration to attributes on the `<dbcp-connection-pool>` element:
 - `maxActive`
 - `maxIdle`
 - `maxWait`
 - `minEvictableIdleTimeMillis`
 - `numTestsPerEvictionRun`
 - `testOnBorrow`
 - `testOnReturn`
 - `testWhileIdle`
 - `timeBetweenEvictionRunsMillis`
 - `whenExhaustedAction`
- If the `driver` attribute of the `<database>` element equals `dbcp` and any of the following attributes are set, the upgrade creates a `<reset-tool-params>` element within the `<dbcp-connection-pool>` element:
 - `collation`

- `oracle.tnsnames`
- `system.username`
- `system.password`

The upgrade then transfers any of these attributes that are defined to the new `<reset-tool-params>` element.

- If the `driver` attribute of the `<database>` element equals `jndi`, the upgrade adds a `<jndi-connection-pool>` element and copies the `jndi.datasource.name` parameter to the `datasource-name` attribute of the `<jndi-connection-pool>` element. If the original configuration did not include a `jndi.datasource.name` parameter, then the upgrade logs an error.
- If the old configuration includes an `<upgrade>` element within the `<database>` element, the upgrade adds an `<upgrade>` element to the `<database>` element of the new configuration.
- If the old configuration contains an `<upgrade>` element that includes an `oracleMarkColumnsUnused` attribute, the upgrade converts the attribute to a `deferDropColumns` attribute, preserving the value.
- If the old configuration contains an `<upgrade>` element that includes a `verifySchema` attribute, the upgrade copies this attribute to `<upgrade>` element of the new configuration.
- If the old configuration contains an `<upgrade>` element that contains an `<oracleddloptions>` or `<sqlserverddlopts>` element, the upgrade logs a warning. You must upgrade these elements manually.
- If the old configuration includes a `<databasestatistics>` element within the `<database>` element, the upgrade copies the `<databasestatistics>` element to the `<database>` element of the new configuration.
- For Oracle databases, if the `<database>` element includes any of the following parameters, the upgrade creates an `<oracle-settings>` element within the `<database>` element of the new configuration:
 - `queryRewriteEnabled`
 - `statisticsLevel`
 - `stored.outlines`
 - `UseDbResourceMgrCancelSql`

The upgrade converts any of the above parameters to attributes on the new `<oracle-settings>` element. The attributes have the following names:

- `query-rewrite`
- `statistics-level-all` (if any value is set for `statisticsLevel` in the old configuration, the upgrade sets the `statistics-level-all` attribute to true in the new configuration. The value `ALL` was the only valid value for the `statisticsLevel` parameter in the old configuration.)
- `stored-outline-category`
- `db-resource-mgr-cancel-sql`
- For SQL Server databases, if the `<database>` element includes either the `msjdbctracelevel` or `msjdbctracefile` parameter, the upgrade adds a `<sqlserver-settings>` element within the `<database>` element of the new configuration. The upgrade then converts the `msjdbctracelevel` and `msjdbctracefile` parameters to `jdbc-trace-level` and `jdbc-trace-file` attributes on the `<sqlserver-settings>` element.
- For SQL Server databases, if the `unicodecolumns` parameter is defined in the old configuration, the upgrade adds a `unicodecolumns` attribute to the `<sqlserver-settings>` element of the new configuration. If the `<sqlserver-settings>` element has not yet been created, the upgrade creates the element.
- If any `<tablespacemapping>` elements are defined in the old configuration, the upgrade creates an `<upgrade>` element within the `<database>` element of the new configuration if one does not yet exist. The upgrade then does the following, depending on the database type:
 - For Oracle, the upgrade adds an `<ora-db-ddl>` element within the `<upgrade>` element of the new configuration, if an `<ora-db-ddl>` element is not yet defined. The upgrade then adds a `<tablespaces>` element to the `<ora-db-ddl>` element and converts each `<tablespacemapping>` element to an attribute on the `<tablespaces>` element. The upgrade then adds an `<ora-lobs>` element to the `<ora-db-ddl>` element and sets the `<ora-lobs>` attribute type to `BASICFILE`.

- For SQL Server, the upgrade adds an `<mssql-db-ddl>` element within the `<upgrade>` element of the new configuration, if an `<mssql-db-ddl>` element is not yet defined. The upgrade then adds a `<mssql-filegroups>` element to the `<mssql-db-ddl>` element and converts each `<tablespacemapping>` element to an attribute on the `<mssql-filegroups>` element.

Changes to Work Queue Configuration

Guidewire has updated the `<work-queue>` element that defines the configuration of a work queue. These changes accommodate improvements to how work queues detect and handle orphaned work items.

The `<work-queue>` element now includes a `retryInterval` attribute. This attribute defines how long in milliseconds to wait before retrying a failed item. The default value is 0, meaning BillingCenter will retry processing the item immediately. During the upgrade, the Configuration Upgrade Tool sets `retryInterval` to 0 if `checkInAfterError` was true, or to the current value of `progressinterval` if `checkInAfterError` was not true.

The `<work-queue>` element also now includes a `retryLimit` attribute. This attribute defines how many times BillingCenter retries a failed work item for the work queue. If you do not specify a `retryLimit` value for a work queue, BillingCenter uses the value of the `WorkItemRetryLimit` configuration parameter in `config.xml` as the default value.

BillingCenter 8.0 no longer includes the `minpollinterval`, `orphansFirst`, or `checkInAfterError` attributes. The Configuration Upgrade Tool removes these attributes.

Changes to Scheduled Purge Batch Processing

Batch processing that purges database records has been added to the default batch processing schedule. These processes include:

- Process History Purge
- Purge Workflows
- Purge Workflow Logs
- WorkItem Set Purge
- Work Queue Instrumentation Purge

Other batch processing that purges database records was already included in the schedule.

During the configuration upgrade, review the updated schedule in `scheduler-config.xml` and make changes as needed.

Changes to Database Statistics Updating

Commands for updating database statistics are now options for the `system_tools` command. If you have custom scripts to schedule database statistics updating, you must modify those scripts.

BillingCenter 7.0

Use `maintenance_tools` with one of the following options:

```
-startprocess dbstatistics  
-startprocess incrementaldbstatistics
```

BillingCenter 8.0

Use `system_tools` with one of the following options:

```
-updatestatistics description false  
-updatestatistics description true  
-cancelupdatestats  
-getupdatestatsstate
```

Specify `true` to update database statistics only for tables exceeding the change threshold. The change threshold is defined by the `incrementalupdatethresholdpercent` attribute of the `databasetrigger` element in `database-config.xml`. Specify `false` to gather full database statistics.

Cancelling Database Statistics Updating

You can now cancel the database statistics updating process while it is running. This capability comes with changes to how you run database statistics updating.

To cancel the process that updates database statistics, use the following command:

```
system_tools -password password -cancelupdatestats
```

In BillingCenter 8.0, the database statistics updating process can be paused just as with other work queues. Use the **Work Queue Info** page to pause an in-progress work queue.

Commands for Database Statistics Update Statements Moved

The `-getdbstatisticsstatements` and `-getincrementaldbstatisticsstatements` commands to get full or incremental statistics updates statements are also now on `system_tools` instead of `maintenance_tools`.

New and Changed in Globalization in 8.0

In BillingCenter 8.0, the *Globalization Guide* replaces the *Localization Guide*. The name change reflects the increased scope of the guide. In earlier releases, the guide described features for localizing BillingCenter to a specific locale. In 8.0, the guide describes features for configuring Guidewire BillingCenter to operate in a global environment.

This topic includes:

- “New to Globalization in BillingCenter 8.0” on page 79
- “Changes to Globalization in BillingCenter 8.0” on page 87

New to Globalization in BillingCenter 8.0

This topic includes:

- “Separation of Language and Regional Format” on page 80
- “Base Configuration Language Support” on page 80
- “Base Configuration Region Support” on page 80
- “Base Configuration Currency Support” on page 81
- “Base Configuration Geographical Support” on page 81
- “Phone Number Data and Phone Validator Gosu Class” on page 82
- “Automatic Address Completion and Fill-in Plugin Functionality” on page 83
- “Ability to Provide a Sort Order for the Typecodes in a Typelist” on page 83
- “Ability to Localize Script Parameter Descriptions” on page 83
- “Language Module Installation Utility” on page 84
- “Globalization-related Configuration Parameters” on page 84
- “Globalization-related Typelists” on page 84

- “Ability to Set First Day of Week by Region” on page 85
- “New in the BillingCenter 8.0 Data Model” on page 85

Separation of Language and Regional Format

New in BillingCenter 8.0 is a separation between language and regional format. As part of the change in how BillingCenter 8.0 provides support for globalization, users can now set their preferences for language and for regional formats independently of each other.

The **International** choice on the **Options** menu provides both **Language** and **Regional Formats** submenus. The **International** choice on the **Options** menu is available only if you configure BillingCenter 8.0 with multiple languages or regional formats.

Base Configuration Language Support

In the base configuration, Guidewire has a single default display language, which is English. If your installation has more than one language, then you can select among the multiple languages using the BillingCenter **Options** → **Language** submenu. BillingCenter uses the **LanguageType** typelist to defines the set of language choices that the **Language** menu shows.

File localization.xml

In 8.0, Guidewire splits file `localization.xml` into multiple files, each of which contains formatting and parsing rules for a single language only. Each base configuration, each `localization` file resides in the following location in Studio:

`configuration` → `Localizations` → `xx_yy`

`xx_yy`, in this case, is a standard ISO locale code.

In a language module, `localization.xml` resides in `config/locale.xx_yy`, with `xx_yy` again being the ISO locale code.

In the PolicyCenter 8.0 release, the existence of a `localization.xml` file in each region is optional. If you do not supply the relevant language-configuration data in a `localization.xml` file, then PolicyCenter uses the ICU library defaults for that language. See “The International Components for Unicode (ICU) Library” on page 21 in the Globalization Guide for details.

See also

- “Working with Languages” on page 23 in the *Globalization Guide*

Base Configuration Region Support

In 8.0, Guidewire provides support for the following core regions in the base configuration:

- United States (English)
- Great Britain (English)
- Australia (English)
- Canada (English)
- Canada (French)
- France (French)
- Germany (German)
- Japan (Japanese)

Regional formats control how BillingCenter formats dates, times, numbers, monetary amounts, and the names of people and companies.

See also

- “Working with Regional Formats” on page 85 in the *Globalization Guide*

Base Configuration Currency Support

In BillingCenter 8.0, Guidewire provides support in the base configuration for the following currencies:

- AUD – Australian dollar
- CAD – Canadian dollar
- EUR – European Union euro
- GBP – British pound
- JPY – Japanese yen
- RUB – Russian ruble
- USD – U.S. dollar

File currencies.xml Split into Multiple currency.xml Files

In the 7.x releases, Guidewire provides a single `currencies.xml` file that contains information on all defined currencies. In 8.0, Guidewire splits the `currencies.xml` file into multiple `currency.xml` files. Each `currency` file contains information on a single currency only.

BillingCenter stores each `currency` file in its own location in Studio, in a separately labeled currency folder:

`configuration → config → currencies → xxx`

For example, the `currency.xml` file for the Australian dollar resides in the following location in Studio:

`configuration → config → currencies → aud`

The effect of this change is to reduce merge conflicts during upgrade.

The following table describes the use of file `currency.xml` in single or multiple currency rendering mode.

Currency rendering mode	Use of currency files
Single currency rendering mode	If operating in single currency rendering mode, then you do not use <code>currency.xml</code> files. Instead, use file <code>localization.xml</code> to provide information for formatting the default application currency.
Multiple currency rendering mode	If operating in multiple currency rendering mode, then BillingCenter ignores file <code>localization.xml</code> . Instead, the BillingCenter installation contains multiple <code>currency</code> files, each of which contains formatting information for a specific currency alone.

See also

- “Configuring Currencies” on page 113 in the *Globalization Guide*

Base Configuration Geographical Support

In BillingCenter 8.0, Guidewire provides support in the base configuration for the following geographical areas:

- AU – Australia
- CA – Canada
- DE – Germany
- FR – France
- GB – Great Britain
- JP – Japan

- US – United States

File address-config.xml Split into Multiple address-config.xml Files

In the 7.x releases, Guidewire provides a single `address-config.xml` file that contains address information for all supported countries. In 8.0, Guidewire splits the single `address-config` file into multiple `address-config` files..

BillingCenter stores the `address-config` files in individual country folders (`xx`) in Studio:

`configuration → config → geodata → xx`

For example, the `address-config.xml` file for the address-related information in Australia resides in the following location in Studio:

`configuration → config → geodata → AU`

See also

- “Configuring Address Information” on page 135 in the *Globalization Guide*

New Multiple country.xml Files

New in 8.0 are `country.xml` files that sets address formatting as it pertains to a single country. For example, you can set which address fields to show, and in what order, for a specific country in this file.

BillingCenter stores separate copies of `country.xml` in its own country folder (`xx`) in Studio:

`configuration → config → geodata → xx`

For example, the `country.xml` file for the address-related information in Australia resides in the following location in Studio:

`configuration → config → geodata → AU`

See also

- “Configuring Address Information” on page 135 in the *Globalization Guide*

File zone-config.xml Split into Multiple zone-config.xml Files

In the 7.x releases, Guidewire provides a single `zone-config.xml` file that contains zone-related information for all supported countries. In 8.0, Guidewire splits the single `zone-config` file into multiple `zone-config` files, each file for use in defining such items as the address autofill fields to use.

BillingCenter stores the `zone-config` files in individual country folders (`xx`) in Studio:

`configuration → config → geodata → xx`

For example, the `zone-config.xml` file for the zone information in Australia resides in the following location in Studio:

`configuration → config → geodata → AU`

See also

- “Configuring Zone Information” on page 126 in the *Globalization Guide*

Phone Number Data and Phone Validator Gosu Class

In 8.0, BillingCenter uses a new Gosu `PhoneValidator` class as the default mechanism to validate phone number correctness.

`gw.api.validation.PhoneValidator`

See also

“Enabling National Field Validation for Phone Data” on page 168 in the *Globalization Guide*

Automatic Address Completion and Fill-in Plugin Functionality

Previous to 8.0, BillingCenter used geographic zone definitions that you loaded into BillingCenter to provide address autofill and autocompletion functionality. In 8.0, BillingCenter provides a new plugin, `IAddressAutocompletePlugin`, that you use to interface to external services for autocomplete data.

In 8.0, to illustrate a common use, you can provide a plugin implementation that autofills an address using data that comes from custom code or from an external service. As you enter a postal code value into the address field, the plugin implementation can detect the postal code value and autofill other address fields based on the entered value.

The default Java implementation class for the plugin, `DefaultAddressAutocompletePlugin`, duplicates the pre-8.0 behavior.

The `AddressAutocompleteUtil` utility class from earlier releases now calls static methods on the Address Auto-Complete plugin, so you do not need to modify your existing code.

Note: Guidewire has configured the new (in 8.0) `GlobalAddressInputSet` PCF files to use the new auto-completion plugin functionality. If you have pre-8.0 PCF files that you use for address entry, then Guidewire strongly recommends that you convert these files to the use of the new `GlobalAddressInputSet` PCF files. See also “Changes to Address Configuration 8.0” on page 87.

See also

- “Address Autocompletion and Autofill” on page 143 in the *Globalization Guide*
- “Automatic Address Completion and Fill-in Plugin” on page 265 in the *Integration Guide*

Ability to Provide a Sort Order for the Typecodes in a Typelist

New in BillingCenter 8.0 is the ability to set the sort order (`priority` value) for the typecodes in a typelist by language. BillingCenter determines the typecode sort order from a file named after the typelist, with a file extension of `.sort`. The typecode `priority` values set in the `.sort` file overrides the typecode `priority` values set in the typekey definition file for that language. BillingCenter stores the sort order information, by language, in the typelist table.

This functionality is useful primarily for configurations that support more than one language, with one of those languages being Japanese. For example, one can use a sort order file to provide a sort order for Japanese provinces, which are customarily ordered from North to South (Hokkaido, Aomori, Iwate, ...).

Guidewire does not provide any sort order files in the base configuration. You must place any `.sort` file that you create in the appropriate configuration folder. For example, for Japanese, place the file in the following location in Studio:

`configuration → config → Localizations → ja_JP`

IMPORTANT Any change that you make to a typelist sort order file triggers a database upgrade.

Ability to Localize Script Parameter Descriptions

New in BillingCenter 8.0, it is possible to localize the descriptions of script parameters. BillingCenter displays script parameter descriptions in the **Administration** tab, **Script Parameters** screen. To localize a script parameter description, you must add a display key for the script parameter to file `display.properties` in the localization

folder for the target language. To see a localized script parameter description on the **Script Parameters** screen for a particular language, users choose that language from the Options  menu.

See also

- “Localizing Script Parameter Descriptions” on page 53 in the *Globalization Guide*

Language Module Installation Utility

New in 8.0 is the ability to install a Guidewire-provided language module using the BillingCenter gwbc utility. You use this utility to install or upgrade an 8.0 language module.

See also

- “Installing Display Languages” on page 25 in the *Globalization Guide*
- “Key BillingCenter gwbc Commands” on page 57 in the *Installation Guide*

Globalization-related Configuration Parameters

The following configuration parameters, defined in config.xml, are new in BillingCenter 8.0:

- DefaultApplicationLanguage
- DefaultPhoneCountryCode
- DefaultNANPACountryCode
- AlwaysShowPhoneWidgetRegionCode
- MulticurrencyDisplayMode

In 8.0, Guidewire has added configuration parameter MulticurrencyDisplayMode that controls whether the application operates in single currency rendering mode or multicurrency rendering mode. The application server does not start if you change the value of MulticurrencyDisplayMode from MULTIPLE to SINGLE. Changing the value of MulticurrencyDisplayMode from SINGLE (the default) to MULTIPLE is irreversible unless you drop the application database.

IMPORTANT If you are integrating multiple Guidewire applications, you must set the value of DefaultApplicationCurrency and MulticurrencyDisplayMode to be the same in each Guidewire application.

See also

- “Globalization Parameters” on page 48 in the *Configuration Guide*

Globalization-related Typelists

The following typelists related to globalization are new or have a new function in BillingCenter 8.0.

Typelist	Description
LocaleType	Use to define supported regions. The value of configuration parameter DefaultApplicationLocale must be a typecode from the LocaleType typelist.
LanguageType	Use to define supported languages. The value of configuration parameter DefaultApplicationLanguage must be a typecode from the LanguageType typelist. Typelist LanguageType exists in 7.x releases. However, in the 7.x releases, the typelist specifies an overloaded set of combined language and locale values. In 8.0, Guidewire constrains the LanguageType typelist to specify a set of language values only.

Ability to Set First Day of Week by Region

Guidewire has added an optional `firstDayOfWeek` attribute to the `<GWLocale>` element in `localization.xml`. You use this attribute to set the day on which a week begins for a particular region. If you do not provide a value for this attribute, then BillingCenter uses the default ICU (International Components for Unicode) value for the region.

The base configuration uses the following ICU first days of the week for the listed regions:

Sunday	<ul style="list-style-type: none">• en_AU• en_CA• en_US• fr_CA• ja_JP
Monday	<ul style="list-style-type: none">• de_DE• en_GB• fr_FR

See also

- For more information how BillingCenter uses the ICU library, see “Locale Codes, `localization.xml`, and the ICU Library” on page 86 in the *Globalization Guide*.

New in the BillingCenter 8.0 Data Model

The following items are new in the BillingCenter 8.0 data model:

- “New Phone Fields on Contact and Person Entities” on page 85
- “New Globalization-related Fields on Contact and Address Entities” on page 85
- “Monetary Amount Data Model Element in BillingCenter 8.0” on page 86
- “Gosu MonetaryAmount Entity in BillingCenter 8.0” on page 86
- “Monetary Amount Widgets in BillingCenter 8.0” on page 86
- “Attributes on `<localization>` Column Element” on page 86

New Phone Fields on Contact and Person Entities

In 8.0, Guidewire has added the following new phone-related fields to `Contact` and `Person` entities:

Entity	New phone-related field
Contact	<ul style="list-style-type: none">• FaxPhoneCountry• HomePhoneCountry
Person	CellPhoneCountry

New Globalization-related Fields on Contact and Address Entities

In 8.0, Guidewire has added the following globalization-related fields to `Contact` and `Address` entities:

- `Contact.NameKanji`
- `Contact.FirstNameKanji`
- `Contact.LastNameKanji`
- `Contact.Particle`
- `Address.AddressLine1Kanji`
- `Address.AddressLine2Kanji`
- `Address.CityKanji`
- `Address.CEDEX`
- `Address.CEDEXBureau`

Monetary Amount Data Model Element in BillingCenter 8.0

In 8.0, Guidewire uses a new data model element called `<monetaryamount>` to represent a specific money amount and its associated currency. You can use the `<monetaryamount>` element anywhere you can use a `<column>` element, for example, on `entity` or `extension` objects.

The `<monetaryamount>` element is different from other data model fields in that it corresponds to two columns in the database:

- The first column represents the numerical amount of money involved, for example 500.
- The second column represents the currency associated with the numerical amount, for example, Canadian dollars.

BillingCenter implements that amount column internally as a standard database column with a datatype of `money` or one of its related derivatives such as `positivemoney` or `negativemoney`, for example. BillingCenter implements the currency type column internally as a typekey of the `Currency` typelist.

Gosu MonetaryAmount Entity in BillingCenter 8.0

New in 8.0 is a Gosu `MonetaryAmount` entity that encapsulates a money amount and a currency. To create a new instance of the `MonetaryAmount` object, pass a `BigDecimal` amount and a `Currency` code to the constructor, for example:

```
new MonetaryAmount(500, Currency.TC_AUD)
```

It is possible to use `MonetaryAmount` objects with Gosu arithmetic operators. However, in any arithmetic operation, the currency types for `MonetaryAmount` objects must match. For example, the following operation throws an exception:

```
(500, USD) + (500, JPY)
```

Monetary Amount Widgets in BillingCenter 8.0

In 8.0, BillingCenter provides several new PCF widgets that work with monetary amounts within the BillingCenter user interface. The following list describes these new widgets

PCF name	Description
<code>MonetaryAmountCell</code>	Use to display <code>MonetaryAmount</code> values in BillingCenter
<code>MonetaryAmountInput</code>	Use to enter <code>MonetaryAmount</code> values in BillingCenter

Attributes on `<localization>` Column Element

New in 8.0, are two new attributes on the `<localization>` subelement on `<column>`. The following list describes each attribute:

Attribute	Type	Description
<code>unique</code>	<code>Boolean</code>	If you set this attribute to <code>true</code> , BillingCenter prohibits duplicate values in the set of value for each language.
<code>nullok</code>	<code>Boolean</code>	<p>If you set this attribute to <code>true</code>, BillingCenter flags missing entries that it finds during a database consistency check.</p> <p>For example, suppose that BillingCenter defines more than one language. In this case, BillingCenter stores values for the default application language in the main database table of the entity. BillingCenter stores the values for the additional languages in a separate localization table.</p> <p>During consistency checks, BillingCenter identifies entries in the main database tables for which no corresponding entries exist in the localization table. Values identified as missing are warnings only. A missing language value does not prevent the database from starting.</p>

Changes to Globalization in BillingCenter 8.0

This topic includes:

- “Changes to Address Configuration 8.0” on page 87
- “Changes to Admin Data Localization” on page 87
- “Changes to 8.0 Typelists” on page 88
- “Changes to Files Used with Globalization” on page 88
- “List Sort Methods Support Optional Locale Sensitive Sorting” on page 90

Changes to Address Configuration 8.0

Previous releases of BillingCenter used a large number of PCF files to render addresses within BillingCenter. Many of these PCF files used one mode for each country.

With BillingCenter 8.0, Guidewire provides three PCF modes that cover the countries/regions that BillingCenter 8.0 supports in the base configuration. Each PCF mode defines the address field order used in that country/region and covers the following:

- The field order for the address-entry edit view
- The field order for the address read-only text view

The following table lists the new modes and the countries/regions that each covers.

PCF mode	Used with...
BigToSmall	Japan
PostcodeBeforeCity	France and Germany
Default	All other countries

ClaimCenter 8.0 changes the PCF files that render addresses and the `AddressOwner` classes to use the new `GlobalAddressInputSet`. Guidewire expressly designed `GlobalAddressInputSet` to be the only address input PCF for use in BillingCenter. Guidewire strongly recommends that you convert any existing, pre-8.0 address entry PCF file to use the new `GlobalAddressInputSet` PCF.

BillingCenter uses new Gosu class `AddressFormatter` to format the address display fields. You can extend `AddressFormatter` to handle address formatting for additional countries.

In addition to the new Gosu `AddressFormatter` class, BillingCenter uses a new Gosu `NameFormatter` class to format names appropriately for internationalization.

See also

- See “Configuring Address Information” on page 135 in the *Globalization Guide* for more information.

Changes to Admin Data Localization

Typically, admin data is customer-defined entities with fields that contain names or phrases that need to be shown in the user’s language. For example, the description and subject fields on an activity pattern are administration data.

You enter localized admin data directly within Guidewire BillingCenter, in a table that is visible at the bottom of the screen. This table contains a row for each enabled language in the application. The row contains, by language, fields for each element on that screen that you can localize. BillingCenter only shows the table if there is more than one defined language in the application.

In previous releases, only a limited number of screens in the base configuration contained the localization table. BillingCenter 8.0 provides the ability to localize a much larger set of administration data. It is also possible for you to configure the application and add or remove localization tables by updating the data model and the appropriate PCF file.

Guidewire builds the localization table as a single PCF. BillingCenter includes this PCF (`LocalizedValuesDV.pcf`) in each parent PCF with localizable data.

Changes to 8.0 Typelists

Guidewire has made the following changes to typelists in 8.0:

Typelist	Changes
Country	In 8.0, Guidewire implements the Country typelist consistently within the Guidewire core applications, including Contact Manager. Compared with previous releases, Guidewire has changed the names of some countries on the typelist and added new country names. The names on the Country typelist reflect the Unicode Common Locale Data Repository (CLDR) country names.
Currency	Contains currency code and similar information for the supported currencies.
JurisdictionType	Although, BillingCenter defines jurisdiction categories in this typelist, BillingCenter 8.0 does not use any of the categories or types. Guidewire intends this typelist for use in future releases.
	<p>See also</p> <ul style="list-style-type: none"> “Configuring Jurisdiction Information” on page 125 in the <i>Globalization Guide</i>
State	<p>Guidewire has modified the base configuration State typelist to include state definitions for the following countries:</p> <ul style="list-style-type: none"> AU CA DE JP US <p>See also</p> <ul style="list-style-type: none"> “Configuring State Information” on page 125 in the <i>Globalization Guide</i>
StateAbbreviation	<p>Guidewire has added the StateAbbreviation typelist in 8.0. This typelist includes abbreviations for states in the following countries:</p> <ul style="list-style-type: none"> Australia Canada Germany United States <p>The purpose of the typelist is to disambiguate abbreviations for state names that exist in multiple countries. For example, the abbreviation “WA” is used in the United States for the state of Washington and in Australia for the state of Western Australia.</p> <p>See also</p> <ul style="list-style-type: none"> “Configuring State Information” on page 125 in the <i>Globalization Guide</i>

See also

- “Configuring Geographic Data” on page 123 in the *Globalization Guide*
- “Working with Regional Formats” on page 85 in the *Globalization Guide*

Changes to Files Used with Globalization

Guidewire has modularized certain files used during localization so that each copy of the file affects a single region, language, regional format preference, or currency only. This is in contrast to previous BillingCenter releases in which a file such as `localization.xml` contained data about more than one locale. In BillingCenter 8.0, Guidewire requires a separate `localization.xml` file to store information about the `en_US` locale and the `fr_FR` locale. As there can potentially be multiple copies of a file, the location of each copy of the file must be unique.

In particular, the following files have changed their name, their functionality, their location, or all three:

File	Changes
• address-config.xml • country.xml • zone-config.xml	Files address-config.xml, country.xml, and zone-config.xml now contain information specific to a single country only. All base configuration copies of these files reside in Studio in the following location, with XX being the country code: configuration → config → geodata → xx Substitute xx with a standard ISO country code. In the base BillingCenter 8.0 release, Guidewire provides support for the following countries: <ul style="list-style-type: none">• AU• CA• DE• FR• GB• JP• US
currencies.xml	Contains currency code and similar information for the supported currencies. In single currency mode, currency is defined in localization.xml files under locale folders. See “Specifying Regional Formats for a Currency in Single Currency Mode” on page 121 in the <i>Globalization Guide</i> . In multiple currency mode, currency is defined in currency.xml files in currency folders, such as usd, located in the currencies folder.
config.xml	Globalization parameters new in BillingCenter 8.0: <ul style="list-style-type: none">• DefaultApplicationLanguage• DefaultPhoneCountryCode• DefaultNANPCCountryCode• AlwaysShowPhoneWidgetRegionCode Globalization parameters with modified functionality in BillingCenter 8.0: <ul style="list-style-type: none">• DefaultApplicationLocale• MulticurrencyDisplayMode See also <ul style="list-style-type: none">• “Globalization Parameters” on page 48 in the <i>Configuration Guide</i>
display.properties	There are now multiple display.properties files, one for each local folder. See “Localizing Display Keys” on page 47 in the <i>Globalization Guide</i> .
fieldvalidators.xml	You define national field validators in fieldvalidators.xml files located in country-specific folders in the fieldvalidators folder. See also <ul style="list-style-type: none">• “Configuring National Field Validation” on page 167 in the <i>Globalization Guide</i>.
language.xml	This file is optional. In the base configuration, Guidewire uses this file to override the ICU default collation settings for the German language. All other base configuration languages use the ICU library default collation settings. See also <ul style="list-style-type: none">• “Searching and Sorting in Configured Languages” on page 160 in the <i>Globalization Guide</i>.

File	Changes
localization.xml	<p>In 8.0, Guidewire splits file <code>localization.xml</code> into multiple files, each of which contains formatting and parsing rules for a single language only.</p>
	<p>Each base configuration <code>localization.xml</code> file resides in the following location in Studio:</p>
	<p><code>configuration → Localizations → xx_yy</code></p>
	<p>Substitute <code>xx_yy</code> with a standard ISO locale code, such as <code>en_US</code>.</p>
	<p>In the BillingCenter 8.0 release, the existence of a <code>localization.xml</code> file for a language is optional. If you do not supply the relevant language-configuration data in a <code>localization.xml</code> file, then BillingCenter uses the ICU library defaults for that language.</p>
	<p>See also</p>
	<ul style="list-style-type: none"> • “Configuring a <code>localization.xml</code> File” on page 87 in the <i>Globalization Guide</i>
nanpa.properties	<p>Area codes defined by the North American Numbering Plan Administration (NANPA). You find this file in Studio in the following location:</p>
	<p><code>configuration → config → phone</code></p>
	<p>See also</p>
	<ul style="list-style-type: none"> • “Working with Phone Configuration Files” on page 155 in the <i>Globalization Guide</i> • “Setting Phone Configuration Parameters” on page 156 in the <i>Globalization Guide</i>

List Sort Methods Support Optional Locale Sensitive Sorting

Some Gosu enhancement methods for sorting collections now support optional locale-sensitive sorting. The new method signatures are on the methods `sort`, `sortDescending`, `orderBy`, `orderByDescending`. See “Sorting Lists or Other Comparable Collections” on page 269 in the *Gosu Reference Guide*.

What's New and Changed in 7.0.0

This part includes information about new and changed features in BillingCenter 7.0. For information about new and changed features in 7.0 maintenance releases, see “Release Notes Archive” on page 141.

New and Changed in BillingCenter 7.0

This document describes the new and changed features in BillingCenter 7.0.

This topic includes:

- “Account Funds Tracking” on page 93
- “Policy-level Billing” on page 94
- “List Bill Accounts” on page 94
- “Payment Instruments” on page 94
- “New Producer Write-Offs Screen” on page 94
- “Agency Bill Suspense Items Screen” on page 94
- “Agency Bill Exception Persistence” on page 95
- “Support for Enhanced Integration with PolicyCenter” on page 95
- “Support for Integration with ContactManager” on page 95

Account Funds Tracking

BillingCenter 7.0 introduces Account Funds Tracking. You can use Funds Tracking to view how funds are used within a single account. See “Account Funds Tracking” on page 241 in the *Application Guide* and see the following sections for more details about:

- Funds tracking overview
- Enabling or disabling funds tracking
- Funds tracking configuration

Policy-level Billing

BillingCenter 7.0 introduces Policy-level Billing. You can now invoice all policies on a direct bill account separately, with one invoice stream per policy. See “Billing Levels” on page 299 in the *Application Guide* for details on:

- Account-level billing
- Policy-level billing
- Switching billing levels

List Bill Accounts

BillingCenter 7.0 introduces List Bill Accounts. You can now have one list bill payer account for many policies owned by other accounts. See “List Bill Accounts” on page 331 in the *Application Guide* for details on:

- Viewing list bill accounts and policies
- Creating and editing list bill accounts
- Account-level charges for list bill accounts
- List bill policies

Payment Instruments

BillingCenter 7.0 introduces payment instruments, which are objects that represent how the insured intends to pay, including payment information such as credit card or bank details. See “Payment Instruments” on page 284 in the *Application Guide*.

New Producer Write-Offs Screen

BillingCenter 7.0 adds the producer Write-Offs screen, which makes it easier to view and work with producer write-offs. See “Producer Write-offs” on page 239 in the *Application Guide* for details on:

- Viewing producer write-offs
- Reversing producer write-offs
- Understanding the write-off amount

Agency Bill Suspense Items Screen

BillingCenter 7.0 adds the Agency Bill Suspense Items screen. This screen shows all of the suspense items across all agency bill payments that have been made by the producer. For details, see “To search for an Agency Bill suspense item” on page 292 in the *Application Guide*. Also, “To release an Agency Bill suspense item for distribution” on page 292 in the *Application Guide*.

The Agency Bill Suspense Items screen enables you to:

- View all suspense items (including released items) across all payments and promises for a producer
- Release one or more suspense items from a single place
- Go to a payment or promise that contains a specific suspense item

Agency Bill Exception Persistence

BillingCenter 7.0 adds agency bill exception persistence. BillingCenter now persists whether or not an invoice item is currently an exception and the date the exception occurred. The new `IInvoiceItem` plugin allows you to configure the behavior of invoice item exceptions. See “Invoice Item Plugin” on page 119. In the base configuration agency bill exception is unchanged.

Support for Enhanced Integration with PolicyCenter

BillingCenter 7.0 offers enhanced support for integration with PolicyCenter. The PolicyCenter payment screen of Submission, Renewal, and Rewrite wizards now provide finer control over billing:

- List bill is included in billing method choices
- PolicyCenter users can specify a default payer that is not owner account
- Billing contact available for direct bill
- PolicyCenter users can select an existing invoice stream or define values for new invoice stream

See “Policy Administration System Integration” on page 399 in the *Application Guide* for details.

Support for Integration with ContactManager

BillingCenter 7.0 adds support for integration of BillingCenter with Guidewire ContactManager, as described in the Guidewire Contact Management Guide. For more information see “Contact Management System Integration” on page 421 in the *Application Guide* for details.

New and Changed in Configuration in 7.0

This topic includes:

- “Changes to the Data Model” on page 97
- “Changes to Guidewire Studio” on page 98
- “Changes to Configuration Parameters” on page 99
- “Changes to Web Services” on page 100
- “Changes Related to PCF Files” on page 101
- “Improvements in Localization” on page 103

Changes to the Data Model

In BillingCenter 7.0, Guidewire has made the following changes to the data model:

- Modifications to the Address Entity
- Modifications to Search Functionality
- Modifications to Associative Arrays

Modifications to the Address Entity

Guidewire has modified the **Address** entity in BillingCenter 7.0.0 by adding a **BatchGeocode** Boolean field. The default is `false`. Guidewire uses this field in conjunction with the **Geocodestatus** field to determine which items to add to the queue of objects for geocoding. If this field is `true` and **Geocodestatus** is set to `none`, then the item needs to be added to the work queue for geocoding.

Modifications to Search Functionality

Guidewire has made changes to search functionality in this release of BillingCenter 7.0:

- Added Conditional-use forceEqMatch Attribute to Criterion Element
- Implemented Configurable Linguistic Searching on SQL Server

Added Conditional-use forceEqMatch Attribute to Criterion Element

In BillingCenter 7.0, Guidewire added a new `forceEqMatchType` attribute to the `<Criterion>` element in `search-config.xml`. The `forceEqMatchType` attribute specifies the name of a Boolean property on the criteria entity:

- If this attribute evaluates to `true`, the criterion uses an `eq` (equality) match.
- If this attribute evaluates to `false`, the criterion uses the `matchType` that the `Criterion` specifies to perform the match.

For example:

```
<Criterion property="StringProperty"
    forceEqMatchType="FlagProperty"
    matchType="startsWith"/>
```

This code uses a `startsWith` match for `StringProperty` unless the `FlagProperty` on the criteria entity is `true`, in which case, the match uses an `eq` match type.

For more information, see “The `<Criterion>` Subelement” on page 319 in the *Configuration Guide*.

Implemented Configurable Linguistic Searching on SQL Server

Guidewire now supports configurable linguistic search on SQL Server, enabling you to configure the case-sensitivity and accent-sensitivity of search results. You configure this by setting the strength of the `LinguisticSearchCollation` element on the `GWLocale` element of the default application locale, as specified in `localization.xml`.

IMPORTANT It is possible for this change to have an upgrade impact on installations with a SQL Server database. See “Setting Linguistic Search Collation” on page 257 in the *Upgrade Guide* for more information.

Modifications to Associative Arrays

It is now possible to set a `hasSetter` and `hasGetter` attribute on each `<link-association>` element of an associative array. For example:

```
<link-association hasGetter="true" hasSetter="true">
    <typelist-map field="TAccountType"/>
</link-association>
```

For these attributes:

- If `hasGetter` is `true`, then you can read the property.
- If `hasSetter` is `true`, then you can update the property.

If you do not specify either of these attributes, then BillingCenter defaults to `hasGetter="true"`.

Changes to Guidewire Studio

In BillingCenter 7.0, Guidewire has made the following changes to Guidewire Studio:

- Modifications to Type System Refresh
- Modifications to the Resources Pane

- Modifications to Tools Options
- Modifications to Dialogs

Modifications to Type System Refresh

If you create or modify an entity or entity extension, then Studio no longer requires a restart to recognize your changes.

- Studio immediately recognizes any entities that you add or any entity extensions (to existing entity fields) that you create.
- Studio immediately generates error messages on any existing references to removed types or entity fields.

Modifications to the Resources Pane

Guidewire has modified the Studio Resources pane in BillingCenter 7.0:

- Guidewire has added a **Localizations** node to the Studio Resources pane.
- Guidewire has removed the **Localizations** tab from the **TypeList** editor. Guidewire has moved this functionality to the **TypeList Localization** editor located under the **Localizations** node.
- Guidewire has moved the **Display Key** editor under the **Localizations** node. Use this editor to manage all display keys and their localization strings.
- Guidewire has renamed the previously existing **Web Services** node to **RPC-Encoded Web Services**. The name change indicates that you use this editor only to work with web services that use Remote Procedure Call (RPC) encoding. See “Changes to Web Services” on page 100 for details.
- Guidewire has added a way to manage WS-I compliant web services through Guidewire Studio. To access, select the **Classes** node in the Studio Resources tree, select a package, then right-click and select **New → WebService Collection**. See “Changes to Web Services” on page 100 for details.

See also

- “TypeList Localization” on page 103
- “Display Key Localization” on page 104

Modifications to Tools Options

In BillingCenter 7.0, Guidewire has added the ability to disable Studio from rescanning the local file system on Studio focus. This provides the ability to manually update and refresh the Studio Resources view at will.

To support the manual refresh functionality, Studio contains the following:

- A Studio toolbar icon  to activate the file refresh manually.
- A **Tools → Options → Version Control → General Settings → Refresh files on focus** option that enables and disables automatic rescanning of the file system on Studio focus.

Modifications to Dialogs

Using the keyboard Escape key (**Esc**) while a confirmation dialog box is open now closes the dialog box. This is the same behavior as clicking **Cancel**.

Changes to Configuration Parameters

Guidewire has removed the following configuration parameters in BillingCenter 7.0.0:

- `IgnoreSoapWebServiceReferenceNamespaceCollisions`

Guidewire has added the following configuration parameters in BillingCenter 7.0.0:

- `GlobalCacheSizeMegabytes`
- `UseSafeBundleForWebServiceOperations`

Guidewire has added the following configuration parameters in BillingCenter 7.0.0 that relate to the domain graph:

- `DomainGraphKnownLinksWithIssues`
- `DomainGraphKnownUnreachableTables`

See also

- “Changes to Web Services” on page 100 for more details on `UseSafeBundleForWebServiceOperations`
- “Application Configuration Parameters” on page 27 in the *Configuration Guide* for a description of these configuration parameters.

Changes to Web Services

In BillingCenter 7.0, Guidewire provides support for the following types of web services:

- RPCE web services or RPC Encoded web services
- WS-I web services, for both server (publishing) and client (consuming) web service APIs

As a consequence, you now work with each web service type differently in Guidewire Studio:

Web service type	Studio
RPC-encoded	The Web Services node in the Studio Resource pane is now called RPC-Encoded Web Services .
WS-I	You now create WS-I web services using Classes → package → New → WebServices Collection .

Note: Although not currently deprecated, Guidewire discourages the use of RPC-Encoded web services.

New Configuration Parameter to Support Backwards Compatibility

Guidewire has added a configuration parameter—`UseSafeBundleForWebServiceOperations`—that changes the behavior of bundle commits in web services published on this server. The default value is `true`.

- If set to `false`, the application ignores entity version conflicts as it commits a bundle.
- If set to `true`, the application detects (and does not ignore) entity version conflicts.

This bundle behavior affects RPC-Encoded web services only.

See the description of this configuration parameter in the BillingCenter Configuration Guide for more details.

See also

- For more information on changes to web services in BillingCenter 7.0.0, see “WS-I Web Services” on page 120.
- For information on how to convert your existing web services to WS-I web services, see “Converting Your Own Published Web Services from RPCE to WS-I” on page 124.
- For an introduction to web services, see “Web Services Introduction” on page 27 in the *Integration Guide*.
- For the list of all built-in web services and whether each one is RPCE or WS-I, see “Reference of All Built-in Web Services” on page 29 in the *Integration Guide*.
- For background about the WS-I standard and its use of the Document Literal encoding, see “Calling Web Services from Gosu” on page 65 in the *Integration Guide*.

Changes Related to PCF Files

In BillingCenter 7.0, Guidewire has made the following change to the PCF editor:

- “New iterator Attribute on Buttons” on page 101
- “New disablePostOnEnter Attribute on Cell” on page 101
- “New onEnter Attribute on WizardStep” on page 101
- “New isDraftOnEnter Attribute” on page 101
- “New Link widgets Menu Command” on page 101
- “New Exposes Tab on ListViewPanel Widget” on page 102
- “Deprecated Unintended ListView ContentCell Configuration” on page 102
- “New Modal Cell and CellRef Widgets” on page 103
- “New CheckedValuesMenuItem Iterator” on page 103

New iterator Attribute on Buttons

In BillingCenter 7.0, Guidewire has added an `iterator` attribute as a required attribute on the following buttons:

- Add and Remove buttons
- Iterator buttons
- Tree buttons

New disablePostOnEnter Attribute on Cell

In BillingCenter 7.0, Guidewire has added a new `disablePostOnEnter` attribute to PCF row cells. You use this attribute to dictate whether a change to an input cell triggers a POST operation to the server. This is a Boolean value (either `true` or `false`.)

- If set to `false`, then BillingCenter computes the Gosu expression defined in the `onChange` attribute as a user enters the PCF page.
- If set to `true`, then BillingCenter does not compute the `onChange` condition and does not perform a POST operation.

BillingCenter only computes the `Onchange` Gosu expression as a user enters the PCF page. If you need to change this behavior, then you need to explicitly refresh or reload the page.

New onEnter Attribute on WizardStep

Guidewire has added the ability to specify an `onEnter` attribute on a `WizardStep` element within a `Wizard` PCF. BillingCenter executes the Gosu in such an attribute anytime that a user enters that step through the BillingCenter interface.

New isDraftOnEnter Attribute

Guidewire has added a new PCF configuration option—`isDraftOnEnter`—to determine, upon entering a wizard, whether BillingCenter highlights unsaved work prior to changes being made.

New Link widgets Menu Command

In BillingCenter 7.0, Guidewire has added a `Link widgets` menu item to the `Page Config` menu in the PCF editor. You use this functionality to connect a source widget (for example, a button) and a target widget (for example, a row iterator) in the PCF editor.

See also

- “Linking Widgets” on page 281 in the *Configuration Guide*

New Exposes Tab on ListViewPanel Widget

It is possible to locate a widget that operates on an iterator in a separate PCF file. This requires that you be able to reference that iterator from the other PCF file. To facilitate explicit iterator configuration in BillingCenter 7.0, Guidewire has added an **Exposes** tab to a number of top-level widgets. The following widgets all have the ability to expose iterators, either iterators defined in its file, or an iterator in an included panel:

- [ListViewPanel](#)
- [ListDetailPanel](#)
- [PanelSet](#)
- [PanelRow](#)
- [RowSet](#)

You use the **Exposes** tab to add or to set values for the following:

- [ExposeIterator](#)
- [ExposeRowTree](#)

`ExposeIterator` contains the following attributes:

Attribute	Optional	Description
<code>applicable</code>	Yes	A Boolean value to indicate whether the panel actually exposes the iterator or not. Only use this if the panel is modal in the rare case that some modes do not define all the iterators.
<code>flags</code>	Yes	A comma separated list of flags that the iterator defines. Any iterator that defines one or more flags needs to be able to expose those flags. You use this attribute to set flags on this iterator.
<code>name</code>	Yes	The ID by which the widget is known outside of the PCF file. This is an optional attribute and it defaults to the widget ID if you do not set it explicitly.
<code>valueType</code>	No	The data type of the iterator.
<code>widget</code>	No	The ID of the widget. This can be an iterator or a list view panel. If it is a <code>ListViewPanel</code> , BillingCenter references the first iterator in the list view panel.

`ExposeRowTree` contains a subset of the `ExposeIterator` attributes.

Deprecated Unintended ListView ContentCell Configuration

In releases prior to BillingCenter 7.0, it was permissible to embed a `DetailViewPanel` under a `ListViewPanel ContentCell` to achieve some level of formatting and layout for the cell content. In BillingCenter 7.0, Guidewire deprecates and disallows the following usages:

Deprecated/Disallowed	Recommended alternative
A <code>DetailViewPanel</code> that is nested in a <code>ListViewPanel ContentCell</code> and that contains a <code>ListViewInput</code> (for example, a <code>ListView</code> inside a <code>ListView</code> cell).	Use a <code>PanelIterator</code> to iterate over a <code>ListViewPanel</code> instead.
A <code>DetailViewPanel</code> that is nested in a <code>ListViewPanel ContentCell</code> and that contains more than one editable input.	Keep any composite <code>ContentCell</code> read-only and edit its content through a <code>Popup</code> .

New Modal Cell and CellRef Widgets

In releases prior to BillingCenter 7.0, it was permissible to create multiple modes for a RowSet (under a ListViewPanel) and reference the desired mode under certain condition through RowSetRef. However, this approach was not very efficient if only one Cell under the Row needed to be different under each mode. This configuration also resulted in duplicate PCF configuration as you needed to re-define all Cell widgets under each mode.

With the release of BillingCenter 7.0, Guidewire supports modal Cell widgets, which allows you to reuse other Cell widgets in the Row under all the modes.

New CheckedValuesMenuItem Iterator

With the release of BillingCenter 7.0, Guidewire has added a CheckedValuesMenuItem iterator. This iterator functions in a similar manner to the AddMenuItemIterator iterator. Use it to perform iteration of a CheckedValuesMenuItem widget.

Improvements in Localization

In BillingCenter 7.0, Guidewire has made the following changes to improve and simplify localization:

- Typelist Localization
- Display Key Localization
- Command Line Export Translation File Import Tool

Typelist Localization

With the release of BillingCenter 7.0, Studio provides a **Typelist Localization** editor to facilitate typelist localization. The **Typelist Localization** editor presents typelists and typecodes in a tree similar to the existing **Entity Names** editor. After you select a typecode from the tree, the editor presents you with an editable table of localized names and descriptions for that typecode, for each locale. Guidewire locates the **Typelist Localization** editor under a new **Localizations** node in the **Resources** tree.

Guidewire stores typelist typekey localizations in `typelist.properties` files, one file for each locale. This modification removes typelist localizations from the typelist definition files (files with extension `.tti`, `.ttx` and `.tix`). Instead, the typekey localizations exist in separate property files in manner similar to the `display.properties` file. For example, Guidewire stores the English (US) typelist localizations in file:

```
.../modules/configuration/config/locale/en_US/typelist.properties
```

Studio stores typelist typekey localizations using the following format for name and description respectively:

```
TypeKey.<typelist>.<typekey> = <localized name>
TypeKeyDescription.<typelist>.<typekey> = <localized description>
```

See also

- “Localizing Typecodes” on page 50 in the *Globalization Guide*

Display Key Localization

In BillingCenter 7.0, as BillingCenter initializes the display key system, it scans all the config/locale directories of all modules for display key property files. Using these files, BillingCenter generates a master list of display keys. BillingCenter uses these values for type information, not for the actual localized display value.

Display Difference Utility Tool

Guidewire provides a display key difference tool that does the following:

- It compares each locale configured on the server against the master list of display keys.
- It generates a file containing the missing key and value.

To generate a display key difference report, run the following command from the application bin directory:

```
gwbc displaykey-diff
```

See also

- “Localizing Display Keys” on page 47 in the *Globalization Guide*

Command Line Export Translation File Import Tool

In BillingCenter 7.0, Guidewire provides a command line tool that you can use to manually import a translation file into Studio. To use this tool, do the following:

1. Navigate to your installation bin directory, for example:

```
BillingCenter/bin
```

2. Run the following command:

```
gwbc import-l10ns -Dimport.file="translation_file" -Dimport.locale=destination_locale
```

The command takes the following parameters:

Dimport.file	Specifies the file that contains the translations. It must be in the same format as an export file from Studio. By default, Studio places the export file in the root of the installation directory: <ul style="list-style-type: none">• If you leave the import translation file in the same location, then you need enter only the name of the file to import.• If you move the translation file to a different location, then you enter an absolute path or a relative path to the file from the root of the installation directory.
Dimport.locale	Specifies the destination locale for the translations. The locale must match a BillingCenter locale that you have defined through Studio, for example: fr_FR or ja_JP.

See also

- “Exporting and Importing String Resources” on page 45 in the *Globalization Guide*

New and Changed in Gosu in 7.0

New in Gosu in 7.0

This topic describes new features in Gosu for BillingCenter 7.0. These include:

- “Gosu Support for Intervals” on page 105
- “Gosu Support for Numeric Literals” on page 105
- “New Null-safe Operators” on page 106
- “New Named Function Arguments and Argument Defaults” on page 106

Gosu Support for Intervals

An interval is a sequence of values of the same type between a given pair of endpoint values. Gosu provides native support for intervals. For instance, the set of integers beginning with 0 and ending with 5 is an integer interval containing the values 0, 1, 2, 3, 4, 5. The Gosu syntax for this is `0..5`. Intervals are particularly useful for concise easy-to-understand `for` loops. Intervals could be a variety of types including numbers, dates, dimensions, and names. You can add custom interval types. In other programming languages, intervals are sometimes called *ranges*. For details, see “Intervals” on page 115 in the *Gosu Reference Guide*.

There is no upgrade impact for this change. You can optionally use intervals for new code if you wish.

Gosu Support for Numeric Literals

Gosu now natively supports numeric literals of the most common numeric types, as well as a binary and hex syntax. Gosu uses the special syntax to infer the type of the object.

For example:

- Gosu infers that the following variable has type `BigInteger` because the right side of the assignment uses a numeric literal “`1bi`”. That numeric literal means “1, as a big integer”

```
var aBigInt = 1bi
```

- Gosu infers that the following variables have type `Float` because the right side of the assignment uses a numeric literal with an "f" after the number.

```
var aFloat = 1f
var anotherFloat = 1.0f
```

For details, see “Numeric Literals” on page 58 in the *Gosu Reference Guide*.

New Null-safe Operators

Gosu has new null-safe operators. In general, these operators check whether an operator is `null`. They return `null` if the value to the left of the operator is `null`, or in the case of the null-safe default operator return some other arbitrary value.

The operators are:

- The null-safe property and method operator (`?.`). See “Null-safe Property Access” on page 87 in the *Gosu Reference Guide*.
- The null-safe default operator (`?:`). See “Null-safe Default Operator” on page 89 in the *Gosu Reference Guide*.
- The null-safe index operator (`?[]`). See “Null-safe Indexing for Arrays, Lists, and Maps” on page 89 in the *Gosu Reference Guide*.
- The null-safe math operators (`?+, ?-, ?*, ?/, and ?%`). See “Null-safe Math Operators” on page 89 in the *Gosu Reference Guide*.

New Named Function Arguments and Argument Defaults

In code that calls functions, you can specify argument names explicitly rather than relying on matching the declaration order of the arguments. This helps make your code more readable. For example, typical method calls might look like the following:

```
someMethod(true, false) // what do those values represent? difficult to tell visually
```

Instead of passing simply a series of one or more comma-separated arguments, pass a colon, then the argument name, then the equals sign, then the value.

For example:

```
someMethod(:redisplay=true, :sendUpdate=false) // easy to read code!
```

Additionally, this feature lets you provide default argument values in function declarations. The function caller can omit that argument. If the function caller passes the argument, the passed-in value overrides any declared default value. To declare a default, follow the argument name with an equals sign and then the value.

For details, see “Gosu Functions” on page 100 in the *Gosu Reference Guide*.

Changed in Gosu in 7.0

This topic describes changes in Gosu for BillingCenter 7.0. These include:

- “Gosu Compiles to Java Virtual Machine Bytecode” on page 107
- “Gosu Syntax Changes (Compile Time Issues)” on page 107
- “Gosu Behavior Changes (Run Time Issues)” on page 112
- “New XML APIs and Improved XSD Support Using ‘XmlElement’” on page 116
- “Guidewire XML Modeler (GX Modeler) Upgraded to Use New XML API” on page 117

Gosu Compiles to Java Virtual Machine Bytecode

Gosu now compiles directly to Java Virtual Machine (JVM) bytecode. For you, the most important thing about this improvement is that Gosu run time performance significantly improved. This affects all Gosu code, including Rules, templates, internal Gosu code, and your own custom Gosu code.

For changes to the syntax or the run time behavior due to this change, see the following sections:

- “Gosu Syntax Changes (Compile Time Issues)” on page 107
- “Gosu Behavior Changes (Run Time Issues)” on page 112

Gosu Syntax Changes (Compile Time Issues)

Because of the change for Gosu to compile directly to Java bytecode, the Gosu compiler and the Gosu run time behavior changed in some ways. The following sections list side effects of this change that change Gosu syntax slightly in ways that some code syntax is no longer supported.

Mismatched property Getter/Setter Types

You can no longer define properties on a Gosu class in which the getter and the setter return different types. For example, do not define both of the following:

- a property getter (`property get` function) that takes an `Integer`
- a property setter (`property set` function) that takes a `String`.

A property can only have one type in Gosu. A property is an abstraction for an underlying field.

Gosu detects this issue at compile time. Gosu flags it as a compile error. Refactor the code to make the types match. To minimize compatibility issues, change the one that was declared second, since previous Gosu releases were ignoring the type of the second mismatched getter or setter.

UPGRADE TASK

Priority: Before starting the server

Summary: Change properties that use different types for getters and setters to use the same types.

Overriding a Generic Function with a non-Generic function

A generic function is a function with a generic declaration in the function like `function foo<T>(arg : T) : T`. For more information about generics, see “Gosu Generics” on page 247 in the *Gosu Reference Guide*.

If you override a generic function, you must override it with a function with exactly the same parameterization. This was previously not a requirement.

Gosu detects this issue at compile time. Gosu flags it as a compile error. To fix it, change the child declaration to exactly match the overridden method declaration.

UPGRADE TASK

Priority: Before starting the server

Summary: Change overridden generic functions that have a different parameterization to match the parameterization of the overridden method declaration.

Covariantly Overriding the Getter Half of a Writable Property

For properties defined as writable on a superclass, you cannot covariantly override getters (`property get` functions) on a subclass. A property must only have one type.

For example, suppose the parent class has a writable property of type `Object` but the child overrides the getter to return the more specific type `String`. You cannot override the setter on the child to take a `String`. Even if you could, you could cause run time errors in other code. For example, other code that had an instance of the object typed as the superclass and tried to set the property to an `Integer`. That could generate an error on the setter call or an error on subsequent getter calls.

Gosu detects this issue at compile time. Gosu flags it as a compile error.

There are two possible fixes:

- You can make the property read-only and define a separate, explicit setter method. You can then override the setter on the subclass to verify the argument type so things at least fail fast with a reasonable error message.
- You can generify the class and have the property be of that type. This approach ensures that the getters and setters continue to have the same type. For example:
 - a. Generify the superclass as `<T>`.
 - b. Define the property as of type `T`.
 - c. In the subclass, declare `MySubclass extends MyParentClass<String>` to ensure the property has the type `String`.

UPGRADE TASK

Priority: Before starting the server

Summary: Change overrides of getters that use different types than the superclass to use the same types.

Overriding an Enhancement Method

Enhancement methods cannot be overridden on a subclass of the enhanced class.

Gosu statically dispatches enhancement methods at compile time. In Java bytecode, enhancements link to actual static methods. As a result, subclasses cannot override them, nor can enhancements override other methods.

Gosu detects this issue at compile time. Gosu flags it as a compile error if you marked the method as an override. Otherwise, Gosu flags this code as a compile warning rather than an error.

To fix this code, put the method on the superclass directly, instead of using an enhancement. If that is impossible, put `instanceof` calls into the enhancement on the superclass to simulate dynamic dispatch.

UPGRADE TASK

Priority: Before starting the server

Summary: Change enhancement methods that override methods defined on superclasses of the enhancements.

Method Overloading Involving Non-Java-backed Types as the Arguments

You cannot overload a method signature in which the only difference is in non-Java-backed types from a type loader. Although this does not necessarily affect all types from a type loader, it affects many types from a type loader. For example, SOAP APIs from an external system or PCF types are non-Java-backed.

Normally, for Java bytecode for the JVM, at run time the JVM looks up the class on disk by its name.

For non-Java-backed types, there may not be a one-to-one correspondence between custom types and Java classes on disk. So, the Gosu compiler finds the common denominator between the custom type and the Java representation, typically `Object`. In some sense, the generated bytecode flattened down to that class, typically `Object`.

For example, means that two methods named `goToPage()` on two different PCF types override compile to `goToPage(PCFElement)`, resulting in duplicate method definitions on the `PCFElement` class.

Gosu detects this issue at compile time. Gosu flags it as a compile error.

There are two options to fixing this issue:

- Create a single method that takes a generic parameter (or even just `Object`) and then dispatch within it by using `typeis` on the argument. Note that `typeis` on a `null` argument always returns `false`, so you must handle `null` arguments specially.
- Put each overloaded variant in a separate enhancement class. Because each enhancement class compiles to its own class, there are no duplicated methods. A downside of this approach is that you cannot override any methods.

UPGRADE TASK

Priority: Before starting the server

Summary: Change overloaded methods that vary by non-Java-backed types as arguments.

Variables With No Type Cannot Initialize to Null

Generally speaking, the type inference feature in Gosu allows you to declare a new variable and omit the type declaration. Gosu uses the type of the object on the right hand side of the equals sign to determine the type of the variable.

However, Gosu now forbids declaring a variable to have the value `null` without declaring the type. In other words, do not write code such as:

```
var x = null
```

A variable has to have a type on the stack. Implicitly or explicitly using the type `void` (technically the type of `null`) does not fix this problem for the Java Virtual Machine. This was previously a compiler warning, but it really indicates a programming error. Gosu now flags this as a compile error.

To fix this, explicitly assign the appropriate type to the variable and do not use the type `void`.

For example, you could change the previous example to:

```
var x : String = null
```

UPGRADE TASK

Priority: Before starting the server

Summary: Add type declarations to variables that you initialize to no value.

Properties Must Not Conflict with Explicit Getter or Setter Methods

Gosu now strictly prevents Gosu properties from creating implicit methods that conflict with methods on a type. Java has no concept of what Gosu calls *properties*. Gosu compiler generates method signatures in Java bytecode from the Gosu properties. This means that Gosu property `get` and `set` functions and regular methods now can collide.

Gosu properties generate implicit get/set methods, depending on the readability and writability of the property. For example, a `Name` property implicitly generates a `getName` and `setName` method, assuming the property is both readable and writable. A method with the explicit declared name `getName` or `setName` would conflict with the generated methods from the parameter if the argument numbers and argument types also match.

Gosu detects this issue at compile time. Gosu flags it as a compile error.

To fix this issue, remove the extra functions (methods), which in typical code are redundant and accidental.

UPGRADE TASK

Priority: Before starting the server

Summary: Remove explicit methods that collide with implicit getter and setter methods for Gosu properties.

Do Not Use the Return Value of a Void Function In an Expression

Functions that return nothing can no longer participate in other expressions. In other words, functions that technically have the return type `void` literally return no value. You cannot call a `void` function and use the result in some expression. The call to the method must be a stand alone Gosu statement. When compiled to Java, the value `void` is not a valid value on the stack, so the expression has an undefined result.

For example, if the function `syncObject` returns no value, the following code is invalid:

```
if (syncObject() > 0) {  
    print("hello?")  
}
```

Gosu detects this issue at compile time. Gosu flags it as a compile error. Typically this indicates a programming error. To fix this issue, change the method call to a stand alone statement.

UPGRADE TASK

Priority: Before starting the server

Summary: Remove void functions from expressions and put them in stand alone statements.

Checked Exceptions Changes in Gosu

Exceptions are wrapped differently in BillingCenter 7.0.

Previously, you could write exception code where the exception's type is undeclared:

```
try {  
    ...  
}  
catch (e) {  
    ...  
}
```

That caught exceptions wrapped as an evaluation exception. The actual exception object was in that object's `Cause` property. The `Cause` property was `null` if the exception has no checked exception associated with it.

Starting in BillingCenter 7.0, that undeclared type syntax is deprecated.

UPGRADE TASK

Priority: Before starting the server

Summary: In try/catch statements, rewrite undeclared exception types to catch type `Exception` or a more specific subtype. Review code in the catch block to ensure it does not unwrap the exception.

Change your code to use the syntax

```
try {  
    ...  
}  
catch (e : Exception) {  
    ...  
}
```

You can alternatively specify a more specific subtype of `Exception`, such as `IOException`.

After making this change in a catch block, carefully review the code in the catch block in case it was trying to do any unwrapping of the exception. For example, by accessing the `Cause` property, which is known as unwrapping the exception. Unwrapping the exception is no longer necessary after you change the catch block to explicitly declare `Exception` or a subtype of `Exception`.

For more information, including why to catch `Exception` and why not to catch the supertype `Throwable`, see “Catching Exceptions” on page 106 in the *Gosu Reference Guide*.

For a related change that affects Java code that interacts with Gosu, see “Catching Exceptions in Java When Gosu throws Exceptions” on page 115.

Higher Precedence for ‘as’ and ‘typeas’ Operators

The precedence of the coercion/conversion operators `as` and `typeas` has increased to match the more conventional priority in other programming languages.

Consider the Gosu expression `1 + 1 as String` with no explicit parentheses.

- In BillingCenter 3.0, Gosu evaluated it as `((1 + 1) as String)`, which evaluates to 2
- In BillingCenter 7.0, Gosu evaluates it as `(1 + (1 as String))`, which evaluates to 11

Gosu detects this issue at compile time. Gosu flags it as a compile error.

To fix this error, explicitly parenthesize the expression in question. A common situation is adding parentheses to any cast to `String` that occurs within an additive `+ expression` that is not already a `String`.

Case-sensitive Language Keywords

Gosu is now more strict with the proper capitalization of Gosu language keywords.

Gosu detects this issue at compile time. Gosu flags it as a compile error.

To fix this, simply fix the capitalization to match the keyword capitalization as indicated in the compile error.

Relative Imports Discouraged, and Now Sometimes Require Fully-Qualified Type Names

In general, to import a type into Gosu, use the `uses` keyword at the top of the file. In some cases, you can omit the fully-qualified name of a type even without the `uses` keyword and the Gosu type system can find the type unambiguously.

One variant of this is called a relative path, which means to use a path relative to the current package of the code that is running. If you use relative paths to refer to a type, in some cases this does not work in Gosu, which causes a run time error.

IMPORTANT It is best to avoid all relative-path imports of types in any new Gosu code.

Gosu detects this issue at compile time. Gosu flags it as a compile error.

To fix this issue, simply fully qualify the type you want to import. For example, in a Gosu class `com.mycompany.MyClass`, suppose you referenced the type `com.mycompany.subpackage.MyUtil` by typing `subpackage.MyUtil`.

UPGRADE TASK

Priority: Before starting the server

Summary: Change imports of types that use relative paths to use fully qualified paths.

Gosu Behavior Changes (Run Time Issues)

Due to the changing Gosu to compile directly to Java bytecode, there were changes in the Gosu compiler and the Gosu run time behavior. The following sections list changes in the run time behavior.

IMPORTANT The following topic describes changes that affect run time behavior that Guidewire Studio does not flag at compile time as compile errors. Review your code for these issues.

Accessing Package-local Java Classes from Gosu classes in the Same Package

You cannot create a package-local Java class.

If you do the following tasks in order, Gosu disallows it:

1. Create a Java class with *no explicit access modifier*. In other words, do not add public or private access modifier. If you omit an access modifier, the default is called *package protected* (*package-protected*).
2. Create a Gosu class in the same package.
3. From the Gosu class, access the Java class by name and access properties or call methods on it.

In Java, a matching package implies a matching package name and a matching class loader. Since the Gosu classes are in a different class loader from the Java classes, the Java Virtual Machine (JVM) considers them different packages. The JVM throws the `IllegalAccessExceptions` exceptions.

WARNING Gosu does not detect this upgrade issue at compile time. Avoid relying on run time errors to find this problem. Review your project for this issue.

To fix this issue, mark the Java class with the `public` modifier.

UPGRADE TASK

Priority: Before starting the server

Summary: Review code for package-local Java classes and how behavior might change.

Standard Arithmetic Operators Are No Longer Null-safe

Arithmetic expressions of reference types (non-primitive types) are no longer null-safe. In BillingCenter 3.0, if either side of an expression was `null`, the entire expression evaluated to `null`. This now causes an null pointer exception.

Although the old behavior is sometimes the intended behavior, it often is a logical error. This change encourages coders to make the null-safe or non-null-safe choice explicitly.

Starting in BillingCenter 7.0, Gosu has more null-safe operators, for example the null-safe addition operator: `?+`. The null-safe arithmetic operators have the same semantics that the simple operator `+` used to have with `null` values. For more details, see “Null-safe Math Operators” on page 89 in the *Gosu Reference Guide*.

At run time, Gosu throws a `null` pointer exception if the object on either side of the operator is `null`. The exception text for this particular issue contains details of how to make an informed decision about how to handle `null` values.

WARNING Gosu does not detect this upgrade issue at compile time. Avoid relying on run time errors to find this problem. Review your project for this issue.

UPGRADE TASK

Priority: Before starting the server

Summary: Review code for changes to null safety of arithmetic operators.

Array Casts

Do not use the `as` keyword to cast an array to a different array type. If you attempt it, Gosu throws an exception at run time.

WARNING Gosu does not detect this upgrade issue at compile time. Avoid relying on run time errors to find this problem. Review your project for this issue.

To fix this, call the `cast` method that exists on all arrays and pass the type of each element. For example, if you want to cast the array to an array of `Integer` objects, pass the type name `Integer` as an argument.

For example:

```
uses java.lang.Integer

var arrayOfInt : Number[] = {3, 4, 5}

// do not directly cast -- it does NOT work!
// var a = arrayOfInt as Integer[]

// cast from Number[] to Integer[]
var a = arrayOfInt.cast(Integer)
```

UPGRADE TASK

Priority: Before starting the server

Summary: Review code for changes to array casts.

Primitive Property Short-Circuiting

In BillingCenter 3.0, property access was null-safe even with primitive properties. For example, if the type of a property is `int`, code similar to `Obj.IntProp` gets the integer property. However, if `Obj` has the value `null`, the expression returns `null` rather than throw a `null` pointer exception. This created the unusual behavior of letting primitive types contain the `null` value at least temporarily within an expression. This is unusual, since `null` properly can be the value only for a reference type (a non-primitive type). In most cases, this worked, especially for `boolean` types, since `null` coerced to `false`, and typically this returned the expected answer.

In BillingCenter 7.0, Gosu is more strict regarding null-safety of operators. The standard period operator for property access and method invocation is not null-safe. If the object to the left of the period has the value `null` and on the right side there is a primitive property, the expression throws a `null` pointer exception.

There are several exceptions:

- The array property called `length`, which short-circuits to the value 0 if a `null` value is on the left side.
- All boolean properties, which always short-circuit to `false` if a `null` value is on the left side

- The Count enhancement property some collection types.

WARNING Gosu does not detect this upgrade issue at compile time. Avoid relying on run time errors to find this problem. Review your project for this issue.

To fix this issue, use the `?.` operator to force null-short-circuiting a property access for primitive properties. If you use the `?.` operator, Gosu uses the default value (0.0, 0, or `false` as appropriate) for a primitive expression instead of throwing a null pointer exception.

IMPORTANT For more information about null-safe property operators, see “Handling Null Values In Expressions” on page 87 in the *Gosu Reference Guide*.

UPGRADE TASK

Priority: Before starting the server

Summary: Review code for changes to primitive property short-circuiting

Class Loading and Initialization Ordering

Due to the changes to make Gosu compile down to Java bytecode, the order which classes load and initialize by the JVM might change. For example, certain classes might load earlier than they did before.

You must never rely on class load ordering for any functionality. For example, do not do something like:

- Have a static initializer for one class called A that puts something in a static hash map on another class B.
- Have another class C use that map in another static initializer on class C.

If suddenly C starts loading before A, there will be run time errors.

WARNING Never rely on class loading ordering for any functionality.

If you relied on class load ordering in previous versions, you might notice errors that are difficult or impossible to debug when the order changes, like in the BillingCenter 7.0.

If you see any null pointer exceptions in the `c1n1t` method on a class, that refers to the internal method that statically initializes a class. Gosu calls this when any code first references the type. That error implies this type of loading and initialization error. The kinds of errors that could occur because of relying on class loading order could be much more subtle and impossible to debug. Be extremely careful not to rely on class loading order.

WARNING Gosu does not detect this upgrade issue at compile time. Avoid relying on run time errors to find this problem. Review your project for this issue.

If you have this type of error, refactor the code so that you do not rely on class load order.

UPGRADE TASK

Priority: Before starting the server

Summary: Review code for changes to class loading.

Arrays of Non-Java-backed Types with ‘typeof’ or ‘TypeSystem.getFromObject(o)’

If an array contains elements of a native Java or Gosu type, including Java classes and Gosu classes, the `typeof` operator and the `TypeSystem.getFromObject(Object)` method work as expected.

However, the behavior is more complex for non-Java-backed types.

Note: For more information about non-Java-backed types, see “Method Overloading Involving Non-Java-backed Types as the Arguments” on page 108.

This issue affects:

- arrays of Guidewire entity types
- arrays of dynamically-generated SOAP (web service) types
- arrays of dynamically-generated XSD types
- arrays of PCF types

In Java bytecode, these types do not directly correspond to the Gosu equivalent. At run time, the type information is unavailable.

For example, the following code is valid because the types are standard types (built-in types and Gosu classes, in this example):

```
uses mycompany.MyClass
var t1 = typeof(new Integer[])
var t2 = typeof(new MyClass[])
```

However, the following code does not work as expected if the array type uses an XSD type:

```
var t3 = typeof(new xsd.myxsd.MyElementType[])
```

In this example, the objects in the package hierarchy return the less useful run time type `XMLElement[]`. This result is semantically incorrect but corresponds to the run time implementation type of the object as seen by the Java Virtual Machine.

UPGRADE TASK

Priority: Before starting the server

Summary: Review code for changes to arrays of non-Java-backed types with `typeof` or `TypeSystem.getFromObject(o)`.

Catching Exceptions in Java When Gosu throws Exceptions

Exceptions are wrapped differently in BillingCenter 7.0. Exceptions thrown from a `throw` statement in Gosu are no longer wrapped in an `EvaluationException` exception. Any Java code that calls into Gosu and expects to catch a wrapped `EvaluationException` must instead change to catch the actual exception type.

For example:

```
function someGosu() {
    throw new IllegalArgumentException("test")
}
```

In previous releases, in Java code you could catch exceptions with the following code

```
try {
    // call code via ReflectUtil, or invoke a method on a Java interface that's implemented in Gosu
    callSomeGosu()
} catch (EvaluationException e) {
    IllegalArgumentException iae = (IllegalArgumentException) e.getCause();
    // Handle illegal argument exception
}
```

In BillingCenter 7.0, that Java `catch` statement does not trigger. A similar issue happens if the `catch` statement caught the exception types `Exception` or `RuntimeException`. In that case, the `catch` statement would trigger, but Gosu would invoke it directly with the `IllegalArgumentException` rather than wrapping it.

This means that it is now better to write Java exception-handling code like:

```
try {
    // call code via ReflectUtil, or invoke a method on a Java interface that's implemented in Gosu
    callSomeGosu()
} catch (IllegalArgumentException e) {
    // Handle the illegal argument exception variable "e"
```

```
}
```

WARNING Gosu does not detect this upgrade issue at compile time. Avoid relying on run time errors to find this problem. Review your project for this issue.

If you have `try/catch` statements in Java code that uses the Guidewire entity libraries, audit your `catch` statements. Ensure your `catch` statements do not assume that exceptions coming from Gosu are wrapped in an `EvaluationException` exception.

For a related change that affects your Gosu code also, see “Checked Exceptions Changes in Gosu” on page 110.

UPGRADE TASK

Priority: Before starting the server

Summary: Review code for changes to Java code that catches exceptions from Gosu.

Private Variables on Superclasses with the Same Name as a Variable on the Subclass

If a parent class has a private variable, a subclass cannot access it.

In BillingCenter 3.0, if a subclass declared a private variable with the same name as the parent class private variable, Gosu was inconsistent about which one to get or set.

In BillingCenter 7.0, the subclass can create its own private variable or non-private variable with the same name as a private variable on the parent class. Gosu treats as separate the superclass and subclass versions of fields with matching names. The subclass cannot access the private variable of the superclass, which is the correct behavior for `private` keyword.

No compile-time semantics have changed, so this problem cannot be detected at compile time. The private variable is a legal symbol in both classes regardless of the runtime behavior. It is solely a run time change such that those fields are now properly separate.

WARNING Gosu does not detect this upgrade issue at compile time. Avoid relying on run time errors to find this problem. Review your project for this issue.

UPGRADE TASK

Priority: Before starting the server

Summary: Review code for changes to private variables on superclasses with the same name as variables on subclasses.

New XML APIs and Improved XSD Support Using ‘XmlElement’

There is a new set of XML APIs based on the root class `XmlElement` instead of `XMLNode`. The main class that represents an XML element is the class `XmlElement`. The most important feature in the new APIs is improved support for XSDs:

- The `XmlElement` APIs handle a wider set of standard-compliant XSDs.
- The `XmlElement` APIs handle namespaces in a more standards-compliant way.
- The `XmlElement` APIs use improved algorithms for adding properties to Gosu objects based on properties and attributes in an XSD.

For more information about the new APIs, see:

- “Manipulating XML Overview” on page 276 in the *Gosu Reference Guide*
- “Introduction to the XML Element in Gosu” on page 279 in the *Gosu Reference Guide*.

You can still use the older APIs that use `XMLNode`, particularly for code you have already written. This documentation refers to as the *legacy XML APIs*. The XML topic still contains documentation for the legacy XML APIs.

For detailed differences between the new XML APIs and the legacy XML APIs, see “Differences Between Legacy XML APIs and Current XML APIs” on page 277 in the *Gosu Reference Guide*.

Backward Compatibility

You can continue to use the legacy `XMLNode` APIs for specific code you already wrote. They are not deprecated. However, for new code, use the newer `XmLElement` APIs.

There is an additional upgrade-specific feature. For backward compatibility only, you can optionally import an XSD into the Gosu type system using the legacy XML system’s XSD loader. To do this, modify the `compatibility-xsd.xml` file. Specifically, copy the file `BillingCenter/modules/p1/config/registry/compatibility-xsd.xml` into the configuration module at the same path. Next, add an entry for your XSD. Set the `namespace` attribute to the Gosu package name of the schema. For example, if the schema is in the package location `example.pkg` and is called `myschema.xsd`, set the namespace to `example.pkg.myschema`.

Guidewire XML Modeler (GX Modeler) Upgraded to Use New XML API

All APIs in the Guidewire XML Modeler (GX modeler) now use the new XML API, whose primary class is `XmLElement`. This might require changes to your code that interacts with an XML model.

See also

- “Manipulating XML Overview” on page 276 in the *Gosu Reference Guide*
- “Introduction to the XML Element in Gosu” on page 279 in the *Gosu Reference Guide*
- “Differences Between Legacy XML APIs and Current XML APIs” on page 277 in the *Gosu Reference Guide*.

Interceptors Removed

Before BillingCenter 7.0, there was a feature called interceptors that allowed you to trigger custom code before or after method invocation. The mechanism was through annotations that implement the `IPreMethodInterceptor`, and `IPostMethodInterceptor` interfaces. This mechanism was removed as part of the change to bytecode-compiled Gosu in BillingCenter 7.0.

UPGRADE TASK

Priority: Before starting the server

Summary: Review code to remove any Gosu interceptors.

New and Changed in Integration in 7.0

New in Integration in 7.0

This topic describes what is new for integration with BillingCenter 7.0. This includes:

- “Payment Instrument API” on page 119
- “Invoice Item Plugin” on page 119
- “Funds Tracking Plugin” on page 120

Payment Instrument API

This new web service, `PaymentInstrumentAPI`, lets an external system create and retrieve information about payment instruments, such as credit cards. The entity type `PaymentInstrumentRecord` has information about a specific payment instrument for an account or a producer. In an integration with PolicyCenter or another policy administration system, BillingCenter is the system of record for payment instrument information.

See “Payment Instrument Web Service APIs (`PaymentInstrumentAPI`)” on page 112 in the *Integration Guide*.

Invoice Item Plugin

This new plugin, `IInvoiceItem`, has three methods:

- `updateExceptionInfo(distItem : BaseDistItem, event : InvoiceItemExceptionEvent)`

This method lets users specify whether to change the exception information for an invoice item that recently had a distribution item executed or reversed.

- `updateExceptionInfo(writeoff : Writeoff, invoiceItem : InvoiceItem, event : InvoiceItemExceptionEvent)`

This method lets users specify whether to change the exception information for an invoice item that recently had a write-off executed or reversed.

- `updateExceptionInfo(oldCommissionAmount : BigDecimal, invoiceItem : InvoiceItem, event : InvoiceItemExceptionEvent)`

This method lets users specify whether to change the exception information for an invoice item that had changes to its primary active commission or gross amount.

See “Invoice Item Exception Information Plugin” on page 182 in the *Integration Guide*.

Funds Tracking Plugin

This new plugin, `IFundsTracking`, lets you integrate your own funds tracking logic with BillingCenter.

See also

- See “Funds Tracking Plugin” on page 198 in the *Integration Guide* – learn about the new funds tracking plugin
- See “Funds Tracking Basics” on page 241 in the *Application Guide* – learn about the new funds tracking feature, which uses the new funds tracking plugin
- See “Allot Funds for Funds Tracking Batch Processing” on page 123 in the *System Administration Guide* – learn about the batch process that uses the funds tracking

Changes in Integration in 7.0

This topic describes what has changed for integration with BillingCenter 7.0. This includes:

- “WS-I Web Services” on page 120
- “Setting Locale in WS-I Requests for Guidewire Servers Only” on page 126
- “Guidewire InsuranceSuite Plugin Implementations are Versioned” on page 127
- “SOAP Implementation Classes and WSDL Packages Include Version” on page 127
- “Escalation and Exception Plugins” on page 128
- “Changes to Messaging System Safe Ordering” on page 128
- “Changes to Messaging System Resync” on page 129
- “Contact Management in BillingCenter” on page 129
- “Changes to Delinquency Processing Customization” on page 130
- “Changes to Billing Instruction Customization” on page 130
- “Changes to Billing Summaries” on page 131
- “Changes to Making Payments” on page 131
- “Changes to the Invoice Assembler Plugin” on page 132
- “Changes to the Invoice and Related Plugins” on page 132
- “Changes to the Policy Period and Policy System Plugins” on page 133

WS-I Web Services

The web services implementation that previously existed uses the SOAP protocol using the Remote Procedure Call (RPC) encoding. This documentation refers to this older style of web services as RPC-Encoded web services or sometimes shortened to RPCE web services.

There is a new standard called WS-I web services. BillingCenter provides a fully WS-I standard-compliant web services layer for both server (publishing) and client (consuming) web service APIs. This is the recommended approach for all new published web services.

If you previously published RPCE web services with the `@WebService` annotation, you can continue to do so. To reduce ambiguity about RPCE web services versus WSI web services, the old annotation name for RPCE web services changed from `@WebService` to `@RPCWebService`.

WARNING RPCE web services are deprecated as of version 8.0. Immediately begin to convert existing RPCE code to WS-I web services.

If you want to create new WS-I web services or convert your RPCE web services to WS-I, use the new annotation `@WsiWebService`. The syntax of the annotation has changed compared to the older RPCE web services. Most notably, the main web service publishing annotation `@WsiWebService` takes no arguments. The RPCE web services used arguments on the annotation to add additional configuration settings, such as required permissions or minimum run level to access the web service. The newer `@WsiWebService` annotation relies on additional annotation types on the published Gosu class, for example `@WsiPermissions` to set the permissions on the service.

For a upgrade-specific information, see the following topics:

- “Converting Your Own Published Web Services from RPCE to WS-I” on page 124
- “Changes for Consuming Web Services in WS-I” on page 125

For full documentation about publishing or consuming web services, be sure to start with the right documentation for your task. See “Finding Documentation for Connecting to Web Services” on page 125.

Some but not all built-in web services converted to WS-I. You connect to WS-I web services differently than you connect to RPCE web services. To learn which built-in web services are RPCE and which are WS-I, see “Reference of All Built-in Web Services” on page 29 in the *Integration Guide*.

Publishing RPCE Web Services Compared to WS-I Web Services

In addition to the differences in the web service publishing annotation itself there are deeper differences between RPCE and WS-I web services.

WARNING RPCE web services are deprecated as of version 8.0. Immediately begin to convert existing RPCE code to WS-I web services.

Behavior of publishing a web service	RPCE web service behavior	WS-I web service behavior
The basic annotation on a class to publish a web service	<p>@RPCWebService</p> <p>IMPORTANT: This is just a new name for the annotation @WebService. The old name still works and refers to RPCE web services, but it is deprecated.</p> <p>This annotation supports many arguments to configure run level, permissions, and local WSDL creation.</p>	<p>@WsiWebService</p> <p>This annotation supports one optional argument. The argument is the namespace for the web service. See “Consuming Web Service Overview” on page 65 in the <i>Integration Guide</i>. If you do not declare the namespace, Gosu uses the default namespace “http://example.com”.</p>
Does BillingCenter automatically generate WSDL files from a running server?	Yes	Yes, although the URL is different. See “Getting WSDL from a Running Server” on page 42 in the <i>Integration Guide</i>
Does BillingCenter automatically generate WSDL files locally if you regenerate the SOAP API files?	Yes	Yes
Does BillingCenter automatically generate JAR files for Java SOAP client code if you regenerate the SOAP API files?	Yes	<p>No, but it is easy to generate with the Java built-in utility <code>wstool</code>. The documentation includes examples. See “Calling a BillingCenter Web Service from Java” on page 59 in the <i>Integration Guide</i>.</p> <p>IMPORTANT: This is an important difference between RPCE web services and WS-I web services.</p>
Can it serialize a Java object as an argument type or return type?	Yes, in limited situations, such as some Java-based types inside the <code>gw.*</code> package namespace.	No
Can it serialize a Guidewire entity as an argument type or return type?	Yes, if the entity is marked in the data model as <code>Exportable</code> .	<p>No</p> <p>IMPORTANT: All published WS-I web services must have no arguments or return types that are Guidewire entity instances</p>
Can it serialize a XSD-based type?	No	Yes, using the <code>XmlElement</code> APIs.
Can it serialize Gosu class instances (sometimes called POGOs: Plain Old Gosu Objects)?	Yes	<p>Yes, however the class must have the following two special qualities:</p> <ul style="list-style-type: none"> • The class must be declared as <code>final</code>, which means it has no subclasses. • The class must have the annotation <code>@WsiExportable</code>. <p>IMPORTANT: Remember to add the <code>final</code> modifier and the <code>@WsiExportable</code> annotation. If you forget, you cannot use that type as a return type or argument type for a WS-I web services.</p>

Behavior of publishing a web service	RPCE web service behavior	WS-I web service behavior
How to specify the minimum run level?	As an argument to the basic web service publishing annotation	The annotation @WsIAvailability
How to specify the required permissions to use the service?	As an argument to the basic web service publishing annotation	The annotation @WsIPermissions
Automatically throw SOAPException exceptions?	Yes	No. Declare the actual exceptions you want thrown. In general this requirement reduces typical WSDL size.
		If you want identical behavior to RPCE web services, add Throws (ServerStateException, PermissionException, BadIdentifierException) onto your Gosu class. However, that set of exceptions probably is not what you want. Carefully consider what exceptions your web service encounters.
Bundle handling Note: a <i>bundle</i> is a container for entity instances. A bundle helps BillingCenter track what changed in a database transaction. See “Bundles and Database Transactions” on page 341 in the <i>Gosu Reference Guide</i> .	RPCE web services put any entity changes into a default bundle, which you manually commit as needed. Additionally, the bundle management is nonstandard. Any changes in a web service default bundle do not trigger message events or respect concurrent data change exception rules.	There is no default bundle for WS-I web services. You define your own bundle. Use the <code>runWithNewBundle</code> API to create a new bundle. See “Running Code in an Entirely New Bundle” on page 351 in the <i>Gosu Reference Guide</i> .
Logging in to the server	<p>There are two ways in the RPCE system to log into a server.</p> <ul style="list-style-type: none"> • For Java clients that used the generated libraries, the standard approach was to use the <code>APILocator</code> utility class to manage the connection and handling connection pooling. • Other client languages (or Java if desired) could connect to the <code>ILoginAPI</code> web service. 	Do not use either the <code>APILocator</code> utility class or the <code>ILoginAPI</code> web service. They do not apply to WS-I web services.
		Instead, each WS-I request embeds necessary authentication and security information in each request.

Behavior of publishing a web service	RPCE web service behavior	WS-I web service behavior
Package name of web services from the SOAP API client perspective	The namespaces are flattened so that all web services appeared in the same package.	<p>The package in which you put your web service implementation class defines the package for the web service from the SOAP API client perspective. The biggest benefit from this change is reducing the chance of namespace collisions in general.</p> <p>For Guidewire InsuranceSuite customers, this makes it much easier to support backwards compatibility with previous (or future) Guidewire application SOAP APIs. The names could match but the package name specifies the application and version number. For example, the structure of the class name might look like:</p> <p><i>MY_PACKAGE_NAME.bc300.MY_CLASS_NAME.</i></p>
Calling a local version of the web service from Gosu	<p>Gosu creates types in the package <code>soap.local.api.APINAME</code></p> <p>All the references were in that package, independent of the package of the implementation class.</p> <p>You must call the following command to rebuild the local files:</p> <p><code>BillingCenter/bin/gwbc regen-soap-local</code></p>	<p>Gosu creates types that use the original package name to avoid namespace collisions.</p> <p>API references in the package <code>wsi.local.ORIGINAL_PACKAGE_NAME</code></p> <p>You must call the following command to rebuild the local files:</p> <p><code>BillingCenter/bin/gwbc regen-wsi-local</code></p>

Converting Your Own Published Web Services from RPCE to WS-I

Refer to the previous table for major differences between RPCE and WS-I web services.

To convert an RPCE web service to WS-I might be a small or large project, depending on the type of service.

For a simple example, suppose all the parameters and return values of the RPCE web service are simple Gosu objects (sometimes called POGOs, Plain Old Gosu Objects). The challenges of converting the web service is probably low. You probably will need to accommodate changes such as the differences in bundle handling.

In contrast, if your RPCE web services use Guidewire entity instances as parameters or return values, you may need major refactoring to accommodate the conversion to WS-I. You must redesign the code to send and receive non-entity data.

IMPORTANT All published WS-I web services must have no arguments or return types that are Guidewire entity instances.

For example, you could change arguments and return types to the following:

- **Gosu class instances** – Write Gosu classes that include the important properties for that integration point. For example, a check printing service might only need the amount and the mailing address, but not any associated metadata or notes about the check. To work with WS-I web services, you must add the `final` keyword on the class, and also add the `@WsiExportable` annotation on the class. For example:

```
package example.wsi.myobjects
uses gw.xml.ws.annotation.WsiExportable

@WsiExportable
final class MyCheckPrintInfo {
    var checkID : String
```

```
    var checkRecipientDisplayName : String  
}
```

- **XML types created with the Guidewire XML modeler** – Use the Guidewire XML modeler tool (also called the *GX modeler*) to generate a model that contains only the desired subset of properties for each entity for each integration point. Then you can import or export XML data using that GX modeler as needed. See “The Guidewire XML (GX) Modeler” on page 310 in the *Gosu Reference Guide*.

Changes for Consuming Web Services in WS-I

The syntax for consuming WS-I web services is different from RPCE (old style) web services.

There are important differences in how this works in BillingCenter 7.0.0.

Note these important differences:

- In Studio, all RPCE web services appear as icons underneath a single icon. In previous releases, this was labelled **Web Services**. In version 7.0.0, the RPCE web services all appear as they used to appear, except that the **Web Services** icon now says **RPCE Web Services**.
- In Studio, WS-I web service resources now exist as a new resource type called a *web service collection*. The location of the web service collection in the package hierarchy defines the package for the types that Gosu creates from the associated WSDL.
- These WS-I web service resources appear in the same resource hierarchy as Gosu classes. For example, navigate to a package in the Gosu classes hierarchy. Then right-click, then choose **New → Web Service Collection**.

For more information on WS-I SOAP web services and web service collections, see “Consuming Web Service Overview” on page 65 in the *Integration Guide*.

Finding Documentation for Connecting to Web Services

For both consuming and publishing web services, the WS-I syntax is different from the RPCE syntax. Be sure to read the correct documentation for the type of technology you are using, as well as whether you are publishing or consuming a web service from BillingCenter.

Type of Web Service	Documentation for publishing web services	Documentation for consuming web services from Gosu
WS-I	“Publishing Web Services” on page 31 in the <i>Integration Guide</i>	“Consuming Web Service Overview” on page 65 in the <i>Integration Guide</i>
RPCE	RPCE web services are deprecated as of version 8.0.	

See also

- For an introduction to web services, see “Web Services Introduction” on page 27 in the *Integration Guide*.
- For the list of all built-in web services and whether each one is RPCE or WS-I, see “Reference of All Built-in Web Services” on page 29 in the *Integration Guide*.
- For background about the WS-I standard and its use of the Document Literal encoding, see “Consuming Web Service Overview” on page 65 in the *Integration Guide*.

Suite Configuration File Overrides URLs to Guidewire Applications

BillingCenter now has a suite configuration file for web service configuration. This only applies to inter-application integrations among Guidewire applications. This file provides a central place to configure (override) the URLs for other Guidewire products in the suite.

For inter-application Guidewire integrations that use WS-I web services, always use the suite configuration file to set the URLs. The suite configuration file is at the path:

`BillingCenter/modules/platform/config/suite-config.xml`

Do not modify that file directly. Copy it to the configuration module and edit it there.

The file looks like the following:

```
<suite-config xmlns="http://guidewire.com/config/suite/suite-config">
  <product name="cc" url="http://localhost:8080/cc"/>
  <product name="pc" url="http://localhost:8180/pc"/>
  <product name="ab" url="http://localhost:8280/ab"/>
  <product name="bc" url="http://localhost:8580/bc"/>
</suite-config>
```

When importing WS-I WSDL, BillingCenter does the following:

1. Checks to see whether there is a WSDL port of element type `<gwwsdl:address>`. If this is found, BillingCenter ignores any other ports on the WSDL for this service.
2. If so, it looks for shortcuts of the syntax `${PRODUCT_NAME_SHORTCUT}`. For example: `${bc}`
3. If that product name shortcut is in the `suite-config.xml` file, BillingCenter substitutes the URL from the XML file to replace the text `${PRODUCT_NAME_SHORTCUT}`. If the product name shortcut is not found, Gosu throws an exception during WSDL parsing.

For web services that Guidewire applications publish, all WSDL documents have the `<gwwsdl:address>` port in the WSDL. The Guidewire application automatically specifies the application that published it using the standard two-letter application shortcut. For example, for BillingCenter the abbreviation is bc. For example:

```
<wsdl:port name="TestServiceSoap11Port" binding="TestServiceSoap11Binding">
  <soap11:address location="http://172.24.11.41:8480/bc/ws/gw/test/TestService/soap11" />
  <gwwsdl:address location=" ${bc} /ws/gw/test/TestService/soap11" />
</wsdl:port>
```

Always use the `suite-config.xml` file to override the URL including the port number to connect to other Guidewire applications using this shortcut system.

The suite configuration file is used only for WS-I web services, not RPCE web services.

Setting Locale in WS-I Requests for Guidewire Servers Only

For WS-I web services hosted on a Guidewire application, you can set (override) the international locale for this web service. See “Setting Locale in a Guidewire Application” on page 78 in the *Integration Guide* and “Setting Locale in a Guidewire Application” on page 78 in the *Integration Guide*.

For integrations among Guidewire applications, if the published web service is a WS-I service, the client application will set the locale as appropriate.

Changes to Locally-Accessed RPCE Web Services

You can call RPCE web services published on the same server as an outgoing (published) SOAP endpoint. This is primarily used for testing your own SOAP APIs or calling BillingCenter SOAP APIs that only exist as SOAP APIs.

Starting in version 7.0.0, before you use the `soap.local.api.*` syntax, you must regenerate the local files related to RPCE services, including after adding, changing, or removing web services. At a command prompt, change the working directory to `BillingCenter/bin` and type the following command:

```
gwbc regen-soap-local
```

IMPORTANT To access local SOAP APIs for RPCE web services, you must call the `gwbc regen-soap-local` script to regenerate local files. Also, after you add, modify, or delete an RPCE web service, call `regen-soap-local` again.

Guidewire InsuranceSuite Plugin Implementations are Versioned

UPGRADE TASK

Priority: Before deploying a production server

Summary: Review changes to the Submission API web service and change your implementations accordingly

If you have more than one Guidewire application, be aware that some of the plugins exist primarily to connect to other Guidewire applications. If you want to use the built-in plugin implementation to connect to other Guidewire applications, ensure that you use the right version of the plugin implementation. The name of the classes are equivalent but are in different packages, which include application version numbers.

The last part of the package name is the Guidewire application two-digit abbreviation, followed by the application version number with no periods. For example, the shortened version of *BillingCenter 3.0.0* is bc300.

For example, the PolicyCenter plugin that connects to BillingCenter 3.0.0 is at the fully-qualified path:

```
gw.plugin.billing.bc300.BCBillingSystemPlugin.
```

IMPORTANT For suite integrations with other Guidewire applications, you must choose the plugin implementation class that matches the version of your applications. Choose the one with the proper version number of the other application (not the current application) in its package name.

For existing customers, this may be an upgrade task to update implementations classes in the Studio Plugins editor. Set the implementation class to the version of your other Guidewire application. If you have questions about this process, contact Guidewire Customer Support.

UPGRADE TASK

Priority: Before your next upgrade

Summary: Update your plugin implementation classes to the version of your Guidewire application.

SOAP Implementation Classes and WSDL Packages Include Version

BillingCenter changed the package of some SOAP API types to include the version number of the application.

There are two types of changes:

Web Service Implementation Classes Are Organized by BillingCenter Version

The web service implementation classes are now organized by the version number of the current application. This makes it easier over time for Guidewire applications to change their API but still maintain backward compatibility with previous versions of web services that still exist in the application.

The full package name includes one component with the Guidewire application two-digit abbreviation followed by its version number with no periods. For example, the shortened version of *BillingCenter 3.0.0* is bc300.

For example, the BillingCenter 7.0.0 version of this API is at the fully qualified path:

```
gw.webservice.policycenter.bc700.BillingSummary
```

In contrast, the BillingCenter 3.0.0 backward compatible version of the API is at the fully qualified path:

```
gw.webservice.policycenter.bc300.BillingSummary
```

WSDL for Guidewire Applications Are Organized by Destination Application Version

WSDL files are files that describe how to use a web service. Starting in BillingCenter 7.0.0, the WSDL files to another Guidewire application are organized by the version of the destination application. Additionally, all WSDL files are encapsulated as web service collections in Studio. Web service collections are the new way to encapsulate remote resources like WSDL and XSD files in Studio. Refer to “Consuming Web Service Overview” on page 65 in the *Integration Guide*.

The full package name includes one component with the Guidewire two-digit abbreviation for the application, followed by its version number with no periods. For example, the shortened version of *BillingCenter 3.0.0* is bc300.

This organization of the WSDL (and web service collections) makes it easier to ensure compatibility with a specific version of the destination application. A remote application’s web services may change between releases. If that happens, you can choose to connect to the older or newer version of that web service API in the destination application.

The files are in the package:

`wsi.remote.APP_ABBREVIATION.APP_ABBREVIATION_AND_VERSION`

UPGRADE TASK

Priority: Before starting the server

Summary: Change references to SOAP and WSDL packages to paths that include version numbers.

Escalation and Exception Plugins

There are several optional exception and escalation plugins. By default, they just call the associated rule sets and perform no other function. Implement your own version of the plugin and register it in Studio if you want something other than the default behavior. The following table lists the exception and escalation plugins:

Name	Plugin interface	The default action
Activity escalation	IActivityEscalationPlugin	Calls the activity escalation rule set
Group exceptions	IGroupExceptionPlugin	Calls the group exception rule set
User exceptions	IUserExceptionPlugin	Calls the user exception rule set

For more details, see “Exception and Escalation Plugins” on page 271 in the *Integration Guide*.

Changes to Messaging System Safe Ordering

Starting in BillingCenter 7.0, the messaging system supports safe-ordering of messages.

Safe ordering is a messaging feature that prevents related messages from sending out of order from each other. This is called *safe order*. Messages are grouped by their related *primary object* for each messaging destination.

Note: For definitions, see “Primary Entity and Primary Object” on page 305 in the *Integration Guide*.

Messages send in creation order with other messages associated with that same primary object for that destination. Any messages associated with a primary object for that destination are called *safe-ordered* messages. The application waits for an acknowledgement before processing the next safe-ordered message for that same primary object for that destination. In other words, delays or errors for that destination blocks further sending of messages for that same destination for that same primary object.

In BillingCenter, there is no default primary entity. However, every messaging destination can set its own *alternative primary entity* to one of three values in the **Messaging** editor:

- If the alternative primary entity is **Account**, messages associated with an account send ordered by account for that messaging destination.
- If the alternative primary entity is **Producer**, messages associated with a producer send ordered by producer for that messaging destination.
- If the alternative primary entity is **PolicyPeriod**, messages associated with a policy period send ordered by producer for that messaging destination.
- If the alternative primary entity is **Contact**, messages associated with a contact send ordered by contact for that messaging destination.

If a message is associated with a primary object for that destination, the message is a *safe-ordered* message.

Not all messages are associated with a primary object. The logic for how and when to send these *non-safe-ordered messages* is different from the logic for safe-ordered messages. Additionally, the behavior varies based on the messaging destination configuration setting called **Strict Mode**. For details, see “Message Ordering and Multi-Threaded Sending” on page 337 in the *Integration Guide*.

Changes to Messaging System Resync

Starting in BillingCenter 7.0, the messaging system supports resynchronizing (resyncing) messages.

In rare cases some messaging integration condition might fail, such as failure to enforce an external validation requirement properly. If this happens, an external system might process one or more BillingCenter messages incorrectly or incompletely. If the destination detects the problem, the external system returns an error. The error must be fixed or there may be synchronization errors with the external system.

However, suppose the administrator fixes the data in BillingCenter and improves any related code. It still might be that the external system has incorrect or incomplete data. BillingCenter provides a programming hook called a *resync event* (a resynchronization event) to recover from such messaging failures.

To trigger this manually, an administrator navigates to the Administration tab, views any unsent messages, clicks on a row, and clicks **Resync**.

You can trigger resync from the administration user interface or programmatically using the **IMessagingToolsAPI** web service.

For details, see “Resynchronizing Messages for a Primary Object” on page 353 in the *Integration Guide*.

Contact Management in BillingCenter

Integrate with an external contact management system by registering a class that implements **IContactSystemPlugin** plugin interface. For demo purposes only, BillingCenter includes the following plugin interface implementation: `gw.plugin.contact.impl.StandAloneContactSystemPlugin`.

The base BillingCenter configuration supports integration with Guidewire ContactManager, which supports the optional Client Data Management module licensed with PolicyCenter (not BillingCenter). If you have a Client Data Management module license and install ContactManager, do not write your own implementation of the **IContactSystemPlugin** plugin interface. Instead, use the plugin implementation `gw.plugin.contact.impl.ContactManagerSystemPlugin`.

Your implementation of the contact system plugin defines the contract between BillingCenter and the code that initiates requests to an external contact management system.

Additionally, there is a new web service for manipulating contacts. To add, update, or remove contacts, use the **IContactAPI WS-I** web service.

See also

- “Contact Integration” on page 395 in the *Integration Guide*
- “Integrating ContactManager with Guidewire Core Applications” on page 45 in the *Contact Management Guide*

Changes to Delinquency Processing Customization

The `onChargesPaid` method of the `IDelinquencyProcessExtensions` plugin takes a list of `ChargePaidFromUnapplied` instances instead of `AbstractChargePaidTxn` instances.

 **UPGRADE TASK**

Priority: Before starting the server

Summary: Change your code that calls or implements the `onChargesPaid` method of the `IDelinquencyProcessExtensions` plugin interface to take a list of `ChargePaidFromUnapplied` instances instead of `AbstractChargePaidTxn` instances.

Changes to Billing Instruction Customization

In earlier versions of BillingCenter, charges were sliced into invoice items by the `BillingInstruction` plugin implementation, combined with `PaymentPlan` instances and methods on the `ChargeExt` extension to the `Charge` entity. BillingCenter replaces this scheme with *charge slicers*.

Specific changes include:

- Removed the `BillingInstruction` plugin methods `generateInvoiceItems(charge:Charge)` and `createSpecialHandlingInvoiceItems(charge: Charge)`.
- Removed the methods on the `ChargeExt` extension to the `Charge` entity `getInvoiceItemType()` and `getInvoiceItemDate()`.

About Charge Slicers

Instead of using the `generateInvoiceItems(charge:Charge)` plugin method as in earlier versions, BillingCenter 7.0.0 uses a charge slicer to slice charges into invoice items. BillingCenter provides three charge slicers entity types:

- `SpecialHandlingChargeSlicer`
- `PaymentPlan`
- `NonPolicyPeriodChargeSlicer`

Charges with a `BillingInstruction` that has special handling are sliced into invoice items by the `SpecialHandlingChargeSlicer` class. This class replaces the `BillingInstruction` plugin method `createSpecialHandlingInvoiceItems`. The default behavior does not change. Only the location where you integrate your own logic changes.

Non-Special Handling for Policy Period Charges by Payment Plans

For non-special handling of `PolicyPeriod` charges that have `ChargePattern.InvoiceTreatment` as `depositandinstallments`, BillingCenter still slices them into invoice items by using the `PaymentPlan`. As In earlier versions of BillingCenter, the payment plan behavior can be customized in the `IPaymentPlan` plugin implementation.

For non-special handling of `PolicyPeriod` charges that have `ChargePattern.InvoiceTreatment` as other than `deposit` and `installments` (`singledeposit`, `singleinstallment`, `onetime`), BillingCenter now also slices them into invoice items with the `PaymentPlan`. The behavior can now be customized in the `IPaymentPlan` plugin. The default behavior does not change. Only the location where you integrate your own logic changes, from the `BillingInstruction` plugin implementation to the `NonPolicyPeriodChargeSlicer` class.

Non-Special Handling for Non-Policy Period Charges by Non-Policy Charge Slicer

For non-special handling of non-policy period charges, BillingCenter slices them into invoice items by using the `NonPolicyPeriodChargeSlicer` class. The default behavior does not change. Only the location where you integrate your own logic changes, from the `BillingInstruction` plugin implementation to the `NonPolicyPeriodChargeSlicer` class.

UPGRADE TASK

Priority: Before starting the server

Summary: Refactor any code that customized the behavior of the `IBillingInstruction` plugin interface and change your implementation accordingly to use the new charge slicers.

Changes to Billing Summaries

The `BillingSummaryAPI` web service has a new method, `retrievePeriodsBilledToAccount`. The method returns open policy periods that the account pays but does not own. This contrasts with the `retrievePeriodsForAccount` method that was available in earlier versions, which returns only open policy periods that the account owns.

For more information, see “Billing Summary Web Service APIs (`BillingSummaryAPI`)” on page 113 in the *Integration Guide*.

Changes to Making Payments

Methods in the `IBillingCenterAPI` web service that make payments moved to a new web service, `PaymentAPI`. In earlier versions of BillingCenter, `IBillingCenterAPI` provided a single method for making payments, which made payments to an account or a policy depending on which parameter was provided. With this release, `PaymentAPI` provides two separate methods for making payments an account or a policy. The methods `makeSuspensePayment` and `makeAccountAdjustment` moved unchanged from `IBillingCenterAPI` to `PaymentAPI`. The method `processPaymentReversal` moved from `IBillingCenterAPI`, and its name changed to `reverseDirectBillPayment`.

For more information, see:

- “Get Invoice Items Methods” on page 106 in the *Integration Guide*
- “Make an Account Balance Adjustment” on page 110 in the *Integration Guide*

UPGRADE TASK

Priority: Before starting the server

Summary: Refactor any code that made payments through the `IBillingCenterAPI` web service and change your implementation accordingly.

Changes to the Invoice Assembler Plugin

New Invocation

The `moveInvoiceItems` method of the `InvoiceUtil` class performs its operation by calling the Invoice Assembler plugin. The `context` argument passed to the plugin method is `TC_API`. Customized instances of the Invoice Assembler plugin must be able to handle this new scenario.

Support for Custom Contexts

In BillingCenter 7.0.5, support is provided to create custom contexts that can be passed to the Invoice Assembler plugin.

To create a new context, add a new typecode to the `InvoiceAssemblerContext` typelist.

Changes to the Invoice and Related Plugins

The following invoice-related plugins change:

- `IInvoice`
- `IInvoiceStream`

See also “Invoice Item Plugin” on page 119.

Changes to the Invoice Plugin

The method `getPaidStatus` is added to the `IInvoice` plugin interface. It takes an `Invoice` instance. BillingCenter uses this new method to determine what to display in the `Paid Status` column on the `Invoices` and what to return from the `Invoice.getPaidStatus()` method used for integration.

Changes to the Invoice Stream plugin

- Dropped the `charge` parameter from the `getInvoiceStreamPeriodicityFor` method. The method signature now is:

```
getInvoiceStreamPeriodicityFor(InvoicePayer payer, PaymentPlan paymentPlan, Periodicity defaultInvoiceStreamPeriodicity) : Periodicity
```
- Changed the method name `getExistingMatchingInvoiceStream` to `getExistingInvoiceStreamFor`. Otherwise, the method signature remains the same.
- Removed the `customizeInvoiceStreamAfterChangeToInvoicingDates` method. This method is not needed, now that invoice streams hold only nullable overrides of the payer's settings. Invoke the `InvoiceStream.updateAfterChangeToPayerInvoicingAttribute` instead, after a payer changes any custom anchor dates. For example, you added twice-per-week invoice days to an account. Invoke `InvoiceStream.updateAfterChangeToPayerInvoicingAttribute` after the twice-per-week invoice days change on the account to update the streams on the account to use the new days.
- Added the `customizeNewInvoiceStream(InvoicePayer payer, InvoiceStream newInvoiceStream)` method. BillingCenter calls this method whenever a new invoice stream is created, such as from the user interface. It is similar to the existing `customizeNewInvoiceStream` method, except it omits the `Charge` parameter.

- Added the `getAnchorDatesForCustomPeriodicity(InvoicePayer payer, Periodicity customPeriodicity) : List<AnchorDate>` method. It returns the payer's anchor dates for custom periodicities. Invoice streams that have custom periodicities and no overriding anchor dates call this method whenever they need to compute their anchor dates. This method is only used for custom periodicities.
-

UPGRADE TASK

Priority: Before starting the server

Summary: Refactor any code that might be affected by changes to the `IInvoice` and `IIInvoiceStream` plugins and change your implementation accordingly.

Changes to the Policy Period and Policy System Plugins

Changes to the Policy Period Plugin

Added the `hasReceivedSufficientPaymentToConfirmPolicyPeriod(PolicyPeriod policyPeriod) : boolean` method to the `IPolicyPeriod` plugin interface. BillingCenter calls this method whenever a payment is applied to the policy period and its `ConfirmationNotificationState` property is `NotifyUponSufficientPayment`. If this method returns `true`, the policy period notifies the policy administration system that the policy period has been confirmed because sufficient payment has been received.

Changes to the Policy System Plugin

Added the `confirmPolicyPeriod(PolicyPeriod policyPeriod, String transactionId)` method to the `IPolicySystemPlugin` interface. BillingCenter calls this method to confirm the policy period whenever sufficient payment has been received.

UPGRADE TASK

Priority: Before deploying a production server

Summary: Review changes to the `IPolicyPeriod` and `IPolicySystem` plugin interfaces and change your implementation accordingly.

New and Changed in System Administration in 7.0

New in System Administration in 7.0

This topic describes what is new for system administration of BillingCenter 7.0. This includes:

- “DCE VM Strongly Recommended for Development Environments” on page 135

DCE VM Strongly Recommended for Development Environments

The Dynamic Code Evolution Virtual Machine (DCE VM) is a modified version of the Java HotSpot Virtual Machine (VM). The DCE VM supports any redefinition of loaded classes at runtime. You can add and remove fields and methods and make changes to the super types of a class using the DCE VM. The DCE VM is an improvement to the HotSpot VM, which only supports updates to method bodies.

Guidewire strongly recommends the use of the DCE VM for development. Guidewire does not support the DCE VM in a production environment. Performance of the Java Virtual Machine might be impacted by the addition of the DCE VM.

Starting in BillingCenter 7.0, Gosu becomes a compiled instead of an interpreted language. In prior releases, Gosu code ran on top of the JVM through an intermediate layer. So, users could change code (and alter execution) at any time, even in a running system. In BillingCenter 7.0, Gosu code is translated to Java bytecode instructions and registered directly with the JVM (in class form) for execution.

This provides two significant benefits. First, Gosu code runs natively on the VM. Native execution is much faster than interpreted execution. Second, Gosu classes can be directly inspected, instrumented, and tuned just like their Java-based equivalents. Development environments, debugging tools, stack traces, memory analysis tools, and many other tools are able to work directly with Gosu classes and treat them as first-class objects.

However, since Gosu classes are now registered directly with the JVM, a new limitation is imposed by JDks. Just like Java classes, Gosu classes cannot be modified once they have been loaded. There are three ways to make a change to a compiled class in a JVM:

- Restart the JVM. This is easy, but requires a full restart.
- Drop the classloader that loaded the class and force a reload from disk. Only supported if the software is designed to do this for special classes and only performs well for small numbers of loaded classes.
- Use JVM hot swapping. This is the option preferred by Guidewire.

Recently, the Java language specification has added a "swap API" to allow VMs to implement class changes while running. This API provides for the ability to reload changed classes on an as-needed, request basis. Development environments, such as Eclipse, WSAD, or Guidewire Studio, can invoke requests against a running VM to ask the VM to reload a new version of a class. Individual vendor VMs are not required to implement this API. VMs can implement a subset of the API, or they can provide a full, general reload facility.

Following is a summary of reload support for the two Guidewire supported JDKs:

IBM JDK

The only types of changes you can reload are:

- Adding or editing (but not deleting) methods
- Adding and deleting classes

New limitations compared to BillingCenter 3.0: Not being able to delete methods, change the method name, or change a method signature

Sun JDK

The only types of changes you can reload are: (limited)

- Editing (but not adding or deleting) methods
- Adding and deleting classes

New limitations compared to BillingCenter 3.0: Not being able to add or delete methods, change the method name, or change a method signature

In addition to these supported VMs, Guidewire is providing a third option, DCE VM applied to the Sun JDK. DCE VM is only available for the Sun JDK and not the IBM JDK. The DCE VM provides full, generalized reload support for all types of class changes. DCEVM 0.2 has been tested and is certified for use with BillingCenter 7.0. This VM is optional but Guidewire strongly recommends using the VM for individual development workstations because it provides automatic support for all forms of Gosu and Java code changes.

Guidewire does not certify use of the DCE VM for shared development environment and integration testing, sandbox, test, staging, and production environments. Reloading is not relevant for these environments. Guidewire certifies use of the DCE VM only for individual development workstations.

If you elect to use the Sun or IBM VM for individual development workstations, without the DCE VM, Studio still attempts to instruct the VM to reload Gosu class changes. In many cases, such as business rules and PCF files, the reloading works fine with these VMs, since the changes that can be made fall within the native reload capabilities. However, in some cases, changes to Gosu code fall outside the reload capabilities of the VM. If this happens, it is necessary to restart the server to reload the change in much the same way that the server is restarted for data model changes. This could lead to confusion since some Gosu changes are reloaded and others are not depending on the JDK.

See also

- “Installing the Dynamic Code Evolution Virtual Machine” on page 40 in the *Installation Guide*
- “Studio and the DCE VM” on page 70 in the *Configuration Guide*
- <http://java.net/projects/dcevm/>

Changed in System Administration in 7.0

This topic describes what has changed for system administration of BillingCenter 7.0. This includes:

- “Updated System Requirements” on page 137
- “Database Configuration Changes” on page 137

Updated System Requirements

Guidewire has updated the system requirements to install BillingCenter. See the *Guidewire Platform Support Matrix* for current system and patch level requirements. The *Guidewire Platform Support Matrix* is available from the Guidewire Resource Portal at <http://guidewire.custhelp.com>.

Guidewire has updated the supported version of the H2 database engine used for the QuickStart development environment configuration. The supported version of H2 uses a different file format than prior supported versions. H2 does not support an upgrade from the old file format to the new format. Guidewire does not support upgrading BillingCenter on H2. You can use the gwbc dev-dropdb command to remove your old BillingCenter H2 database before you upgrade. BillingCenter creates a new H2 database if one is not found. Alternatively, you can change the jdbcURL in config.xml to point to an unused location for H2 files.

If you attempt to use BillingCenter 7.0 with an old H2 database, you will encounter an error similar to the following:

```
Caused by: org.h2.jdbc.JdbcSQLException: Unsupported database file
version or invalid file header in file "Old database:
C:\tmp\guidewire\bc.data.db - please convert the database to a
SQL script and re-create it." [90048-147]
```

Database Configuration Changes

The maxWait and stmtpool.enabled parameters are removed from the database configuration in BillingCenter 7.0.

The stmtpool.enabled parameter has been removed completely. This parameter enabled Oracle statement caching and statement pooling when set to true. Guidewire has found that enabling statement caching and pooling leads to issues with too many open cursors.

The maxWait parameter has been removed from the default config.xml file. This parameter defines the maximum time in milliseconds that the data source waits for a connection before one becomes available in the pool to service. The default value of 30000 is usually sufficient. You can add the maxWait parameter to the <database> element in config.xml.

New and Changed in Rules in 7.0

This topic covers what is new and changed with the BillingCenter business rules BillingCenter 7.0.

This topic includes:

- “Changes in Rules in BillingCenter 7.0” on page 139

Changes in Rules in BillingCenter 7.0

This topic describes what has changed with rules in BillingCenter 7.0.

There are no changes to rules in this release.

Release Notes Archive

This section contains the release notes for previous versions of BillingCenter. Use these files to learn what features changed from one release to another.

IMPORTANT This section contains upgrade information originally provided for earlier BillingCenter releases. It may be superseded by later release notes or other upgrade documentation.



chapter 17

Guidewire BillingCenter 8.0.3 Release Notes

Release 8.0.3

Document Revision: 02-July-2015

Copyright © 2001-2015 Guidewire Software, Inc. All rights reserved.

Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire Reinsurance Management, Guidewire ContactManager, Guidewire Vendor Data Management, Guidewire Client Data Management, Guidewire Rating Management, Guidewire InsuranceSuite, Guidewire ContactCenter, Guidewire Studio, Guidewire Product Designer, Guidewire Live, Guidewire DataHub, Guidewire InfoCenter, Guidewire Standard Reporting, Guidewire ExampleCenter, Guidewire Account Manager Portal, Guidewire Claim Portal, Guidewire Policyholder Portal, Gosu, Deliver Insurance Your Way, and the Guidewire logo are trademarks, service marks, or registered trademarks of Guidewire Software, Inc. in the United States and/or other countries.

All other trademarks are the property of their respective owners.

This material is confidential and proprietary to Guidewire and subject to the confidentiality terms in the applicable license agreement and/or separate nondisclosure agreement.

Guidewire products are protected by one or more United States patents.

Overview

These release notes contain the following topics:

- Release Information
- Installing This Release
- Support
- Major Issues and Changes
- Improvements and Resolved Issues
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire BillingCenter.

IMPORTANT If you skipped one or more upgrade releases to BillingCenter, be sure to read the Release Notes for those releases to learn about changes and fixed issues.

Version Number

This release of Guidewire BillingCenter is 8.0.3.

- If you are integrating with Guidewire PolicyCenter 7, this version of BillingCenter requires PolicyCenter 7.0.4 or later. The latest maintenance release is preferred.
- If you are integrating with Guidewire PolicyCenter 8, this version of BillingCenter requires PolicyCenter 8.0.3 or later. The latest maintenance release is preferred.
- If you are integrating with Guidewire ContactManager, this version of BillingCenter requires ContactManager 7.0.0 or later. The latest maintenance release is preferred.

Installing This Release

Refer to the following:

- The *BillingCenter Installation Guide* for general installation information
- Prior BillingCenter release notes for any versions that you have skipped

Upgrade Information

Upgrading from BillingCenter 3.0.4 or newer versions is supported in this release.

This topic contains the following:

- Incorrect warning message when running Configuration Upgrade Tool (PL-28723)

Incorrect warning message when running Configuration Upgrade Tool (PL-28723)

When upgrading from one BillingCenter 8 release to another BillingCenter 8 release, running the Configuration Upgrade Tool may produce the following warning message:

```
WARN cannot find Emerald base configuration zip, this could indicate a problem...
```

If the `BillingCenter/modules/base.zip` file does exist, then you can safely ignore this warning.

Support

For assistance, visit the Guidewire Resource Portal – <http://guidewire.custhelp.com>

Major Issues and Changes

This topic contains major issues and changes that can affect your installation.

- New and Changed Features
- Improvements to List View Navigation (PL-26518)

- Base PCF File Changes
- Changes in This Release Provided in Upgrade Diff Report

New and Changed Features

For information on new features and other changes, see the topic “New and Changed in BillingCenter 8.0.3” in the *BillingCenter New and Changed Guide*.

Improvements to List View Navigation (PL-26518)

The following improvements have been made to the behavior of selecting and navigating within a list view:

- Selection now favors selecting a cell rather than a row. A row is still selected where required, such as when selecting a row in a list detail view.
- Use the arrow keys, Tab (next cell), and Shift+Tab (previous cell) to navigate within in a list view.
- With an editable cell selected, press Enter to begin editing it. To leave edit mode, press Esc or click outside of the list view.
- When editing a list view cell, press Tab or Shift+Tab to remain in edit mode and navigate to the next or previous editable cell.
- For cells that have special controls, use standard keyboard actions to change the cell value. For example, press Space or Enter to select a check box or radio button, press Down Arrow to activate a drop-down list, and so on.

Base PCF File Changes

The following link requires that the `ReleaseNotes_files` directory be on your local disk in the same directory as this release notes file.

BillingCenter release 8.0.2 to 8.0.3

To view a report of the changes to the base PCF files, *[click here](#)*.

Changes in This Release Provided in Upgrade Diff Report

Guidewire provides a report detailing certain differences between the current release and your prior release. This report describes changes in display keys, entities, typelists, and the Gosu API. To obtain your custom Upgrade Diff Report, visit the Guidewire Resource Portal at <http://guidewire.custhelp.com>.

Improvements and Resolved Issues

This topic describes improvements and issues corrected in this release. Guidewire attempts to provide information for issues of primary importance to our customers. This list is not intended to be comprehensive.

This topic includes:

- Application Improvements and Resolved Issues
- Platform Improvements and Resolved Issues
- Documentation Improvements and Resolved Issues

Application Improvements and Resolved Issues

ID	Description
Agency Bill	
BC-13421	The Agency Bill Open Items page exhibits improved performance for producers with a large number of policies.
Commission / Commission Plan	
BC-9707	Added consistency checks for extraneous ZeroCmsnEarnedMarkers.
Database Upgrader	
BC-13770	When upgrading from BillingCenter 3 or BillingCenter 7, new BillingCenter 8 permissions are automatically added to the Superuser and General Admin roles. In earlier releases, the new permissions had to be added manually.
BC-14327	Improved the performance of InvoiceItemHasBilledAndDueTxnTrigger.
BC-14328	Improved the performance of AddStatementInvoiceAllInvoiceItemsExactlyPaidVersionTrigger.
BC-14389	Data created by multiple PoITransferByProdCode transfers between the same Producer Codes can be upgraded. In previous releases, data of this type threw a version-check error when upgrading.
General	
BC-12723	The PolicyPeriod.PolicyNumberLong field is constructed by concatenating the PolicyPeriod.PolicyNumber and PolicyPeriod.TermNumber string fields. If the PolicyNumber field is subsequently modified, the PolicyNumberLong field is appropriately updated. In earlier releases, the PolicyNumberLong field was not updated.
BC-12731	Phone numbers retrieved from ContactManager are displayed with appropriate formatting. In previous versions, the numbers were shown sequentially, without formatting.
BC-12891	In BillingCenter versions prior to 8.0, payments could be applied to invoice items of the opposite sign. This capability was removed in BillingCenter 8.0, but existing data created in such a manner can still enter BillingCenter during the upgrade process. Such data is restricted, however, in that it cannot be redistributed when making a change to the Payment Plan using the Reverse and Redistribute option. The Payment Schedule screen in the BillingCenter base configuration does not allow redistribution of the restricted payments. If they exist, a localized warning message is shown summarizing the restricted payments. Developers can identify and retrieve the restricted payments by calling the new getPaymentsThatWontBeRedistributed method in the PaymentPlanChanger class. The restricted payments returned by the method can then be handled in any appropriate manner.
BC-13808	When entering an account number in a search field, leading and trailing spaces are ignored. In earlier releases, extra spaces could cause an exception.
BC-13778	When attaching an account document to an email, the operation succeeds. In earlier releases, the operation would throw an UnsupportedOperationException if the IDocumentMetadataSource plugin was disabled.
BC-13791	Modifications to an executed payment or credit distribution may not be performed directly. For example, an executed distribution cannot be modified by calling the methods of the BaseDistImpl class to alter individual components of the distribution. Attempts to directly change executed payment or credit distributions throw an exception. To change an executed distribution, use the modification framework.
BC-14058	To successfully reverse a suspense payment item, a corresponding unapplied-suspense-payment transaction or suspense-release transaction must exist. If such a transaction does not exist, the reverse operation aborts and throws an IllegalStateException. In previous releases, BillingCenter incorrectly allowed the reversal of any suspense payment item.
BC-14209	When searching for a transaction, the minimum amount and maximum amount fields can specify negative values. Previous releases supported positive values only.
BC-14339	Fixed an issue where incremental updates could be duplicated to denorm values if the bundle was rolled back late in the commit cycle.
BC-14434	Fixed an issue with the DateTime widget. When entering a date with leading zeros (01/01/2014) and leaving the time empty, the widget correctly auto-fills the time to 12:00 a.m. In earlier releases, the time is not auto-filled.

ID	Description
BC-14464	Fixed an issue where dates containing leading zeroes in the month and day were not correctly formatted and interpreted.
Internationalization	
BC-13113	In the Account History screen, the Account Created entry in the Items column is translated to the language of the default locale. In previous versions, the text was not translated.
Invoicing	
BC-13822	When an invoice is carried forward, its items are placed on a planned invoice within the same invoice stream. In earlier versions, the items might have been placed on a planned invoice within any of the account's invoice streams.
BC-14398	<p>In all versions of BillingCenter, an invoice number is generated whenever an invoice is created. When creating a large number of invoices, such as during multiple simultaneous policy-preview operations, the generation of invoice numbers can slow BillingCenter's operation. A performance gain can be achieved by eliminating unnecessary invoice number generation and, instead, generating a number only when an invoice is committed.</p> <p>To generate invoice numbers for committed invoices only, set the config.xml parameter <code>AllocateInvoiceNumberOnInit</code> to <code>false</code>. The default setting is <code>true</code>, which generates an invoice number whenever an invoice is created.</p>
Payment	
BC-14051	When modifying a direct bill or agency bill distribution, the contents of extension fields added to the <code>DirectBillMoneyRcvd</code> or <code>AgencyBillMoneyRcvd</code> entity were incorrectly cleared. This issue is fixed.
Payment Plan and Invoicing	
BC-13738	Fixed the recalculation of invoices when changing the Payment Plan to a more-frequent periodicity. In previous versions of BillingCenter 8.0, the recalculated invoice dates could overlap with billed invoices from the original Payment Plan.
Performance	
BC-14025	<p>Achieved a performance gain when moving from either the Account Charges screen (<code>AccountDetailCharges.pcf</code>) or the Policy Charges screen (<code>PolicyDetailCharges.pcf</code>) to the Transaction screen (<code>TransactionDetailPopup.pcf</code>). The gain was accomplished by eliminating the need to pre-load a large amount of transaction data.</p> <p>The new PCF file <code>ChargeTransactionDetailsForward.pcf</code> was introduced as part of the improvement.</p>
PolicyCenter Integration	
BC-13820	When communicating with PolicyCenter, send only payment instrument types that are supported by PolicyCenter. In earlier releases, all of the account's payment instruments were sent.
BC-13825	When a policy is rewritten in PolicyCenter, BillingCenter recognizes if a different payment plan was selected for the policy. In BillingCenter 8.0.2, the original payment plan would be retained if the communication with PolicyCenter occurred through the bc700 API.
Policy Access Subsystem	
BC-13721	When a user who does not have full security access performs a policy search by entering a policy number, correct search results occur. In earlier releases, the policy number was not accurately utilized in the search operation, causing incorrect results.
Suspense Payment	
BC-14035	Situations existed in previous releases that allowed a suspense payment to be applied multiple times. A suspense payment can now be applied only if a non-reversed Suspense Payment transaction exists.
BC-14045	Fixed an issue that enabled reversed suspense payments to be applied to an account, policy, or producer.
BC-14047	Fixed an issue where reversing a suspense payment that had been edited did not reverse all relevant transactions.
BC-14065	While editing a payment, an associated suspense item can be modified and released. In earlier versions, modifying and releasing a suspense item while editing a payment would create incorrect data.

ID	Description
Trouble Tickets	
BC-13690	<p>Trouble tickets can place a disbursement hold or a payment distribution hold on an individual policy. In earlier versions, such holds could be placed only on an account.</p> <p>For the policy-based hold to have an effect in the base configuration, the account must be using policy-level billing with cash management. If the account uses any other billing level, the policy-based hold can still be placed, but it will have no effect without custom configuration.</p>

Platform Improvements and Resolved Issues

ID	Description
Authentication	
PL-30775	Removed the unused configuration parameter <code>ShouldSyncUserRolesInLDAP</code> .
Build Infrastructure	
PL-27679	The <code>gwbc regen-java-api</code> command now generates Java API Javadoc into the <code>java-api/doc/api</code> directory.
Command Line Tools	
PL-27608	<p>Removed extraneous files from the output of the command <code>zip-changed-config</code>. The list used is in <code>com.guidewire.tools.upgrade2.merge.ConfigMergeList#EXCLUDE_DIRS</code>.</p> <p>Also added new option <code>-e</code>, which enables you to specify a semicolon-separated list of directories to exclude.</p>
Consistency Checker	
PL-30098	Fixed an issue where Database Consistency Checks that checked date parameters would compare against an incorrect date.
Database Support - DB2	
PL-30469	Replaced parameter <code>identifyQueriesViaComments</code> in <code>config.xml</code> with new parameters <code>IdentifyQueryBuilderViaComments</code> and <code>IdentifyORMLayerViaComments</code> . The <code>IdentifyQueryBuilderViaComment</code> parameter instruments high-level database objects constructed by using the query builder APIs. <code>IdentifyQueryBuilderViaComment</code> is set to <code>true</code> in the base configuration. The <code>IdentifyORMLayerViaComments</code> parameter instruments lower level objects, such as beans, typelists, and other database building blocks. <code>IdentifyORMLayerViaComments</code> is set to <code>false</code> in the base configuration.
Database Support - Oracle	
PL-29153	The AWR report will now only report activity for the targeted schema (Guidewire database) on an Oracle server hosting multiple schemas.
PL-29868	The configuration of Oracle range and hash partitioned indexes has changed from the last release. If you have any indexes configured for Oracle partitioning, read the topics on partitioning under "Configuring the Database" in the <i>Installation Guide</i> and make changes accordingly.
PL-30770	Updated upgrade options in <code>database-config.xml</code> files to show default values. Removed an unneeded setting.
PL-31513	Fixed an issue where Oracle LOBs were mistakenly set with the <code>nologging</code> attribute.
PL-31548	If the AWR report experiences an XML parsing error while reading from <code>DBMS_SQLTUNE.REPORT_SQL_MONITOR</code> , the read of the data will be skipped and an explanatory message printed in the Errors section of the report.
PL-31810	Fixed a problem with Guidewire AWR download when the database is Oracle 12.1.0.2 patchset or higher.
Database Support - SQL Server	
PL-30578	Two pages have been added to the SQL Server DMV performance report that aggregate active session statistics by user and by action.

ID	Description
Document Management	
PL-31555	Fixed an issue so that you can now use asynchronous document storage (<code>AsyncDocumentContentSource</code>) even if the Plugins registry has an enabled implementation for the <code>IDocumentMetadataDataSource</code> plugin interface.
Email	
PL-27557	Added an additional constructor in <code>EmailTemplateSearchCriteria</code> to add a callback interface and set a variable when it does not exist in the current evaluation context. This is to cover cases where the environment in which the templates are retrieved (filtered) and the environment in which they are evaluated are not the same.
Entities/Metadata	
PL-30357	Fixed an issue where the <code>gwbc regen-java-api</code> command would throw an exception when processing a typelist that had more than 4,000 typecodes. Now 8,000 typecodes are supported.
External Entities	
PL-28992	Resolved an issue that caused the <code>gwbc regen-java-api</code> command to sometimes fail if the flag <code>-Ddeprecated=true</code> was specified.
Gosu	
PL-29715	Fixed an issue that could cause runtime exceptions if you set the <code>config.xml</code> parameter <code>WarnOnImplicitCoercion</code> to the value <code>false</code> .
IntelliJ IDE - Compiler	
PL-29534	Compilation errors for a widget template are now shown against the containing PCF file rather than against the template.
PL-30918	Fixed an exception that would occur when a PCF file contained a widget template that contained errors.
IntelliJ IDE - Entity Editor	
PL-29124	Fixed an issue where comment elements such as <code><!-- Comment --></code> inside an entity or typelist definition file were removed when edited in Studio.
IntelliJ IDE - Localization	
PL-30209	You can now switch the language in Studio without needing to restart. To do so, navigate to <code>File → Settings → Guidewire Studio</code> , and then under <code>Language Settings</code> , click the language.
IntelliJ IDE - PCF Editor	
PL-30102	Removed a non-functional link in the pop-up help text when hovering over a PCF widget in Studio.
IntelliJ IDE - Plugins	
PL-29776	If you configure Guidewire Studio to work with IDEA IntelliJ Ultimate, now the OSGi Plugin Editor also uses Ultimate.
IntelliJ IDE - Product Model Editor	
PL-29996	Fixed an issue that prevented creating Gosu enhancements for product model coverages.
IntelliJ IDE - Webservices Editor	
PL-28462	Fixed an issue with the Web Services Collection Editor in Studio where it was unable to download imported schemas correctly.
PL-29920	Fixed an error that would occur in the Web Services editor in Studio when clicking Fetch Updates when the <code>suite-config.xml</code> file contained multiple values for the <code>env</code> attribute.
Messaging	
PL-27870	For messaging use, there are Gosu APIs that provide a <code>HashMap</code> that exists across multiple rule set executions. The APIs are methods on the result object of Gosu expression <code>messageContext.SessionMarker</code> . Older methods <code>addToTempMap</code> and <code>getFromGetMap</code> manipulate a <code>HashMap</code> that exists uniquely for each messaging destination. For a hash map that exists across all rules for all destinations, use the new methods <code>addToSessionMap</code> and <code>getFromSessionMap</code> .
PL-28769	Destination ID, message ID, sender ID, and requester user name of retried and skipped messages are now logged.

ID	Description
PL-28952	On Oracle with this release, BillingCenter includes Oracle SQL hints in queries that select messages for a message destination. Oracle SQL hints can improve query performance. You can use the new parameter <code>UseOracleHintsOnMessageQueries</code> in <code>config.xml</code> to enable and disable the use of Oracle SQL hints at run time.
PL-29139	Several changes for inbound integration APIs: <ul style="list-style-type: none"> The <code>InboundIntegrationPlugin</code> plugin interface is now called <code>InboundIntegrationStartablePlugin</code> For messaging use, the new plugin interface <code>InboundIntegrationMessageReply</code> shares features of the inbound integration plugin but is a message reply plugin not a startable plugin Existing file and JMS integrations are now available also as <code>MessageReply</code> variants. See the <i>Integration Guide</i> in the "Multi-threaded Inbound Integration" chapter.
PL-30119	Fixed an issue with the JMS inbound integration APIs that caused the exception "JMS Integration Service not supported on server runtime".
PL-30689	The application no longer performs redundant locking and unlocking of entities if a message destination does not use the optional <code>MessageRequest</code> plugin.
Other - Developer IDE Pod	
PL-27489	Reduced the number of unnecessary library files that <code>regen-java-api</code> generates into the <code>lib</code> folder.
Other - Integration Pod	
PL-27845	Custom implementations of the inbound integration plugin interface (<code>InboundIntegrationPlugin</code>) now support Gosu implementations.
PL-30148	For use with the <code>DataChangeAPI</code> web service, new Gosu APIs for your data change code configure logging to the data change user interface. To log field-level entity changes, call <code>DataChange.util.setDetailResultWriting(bundle)</code> . To log arbitrary text, call <code>DataChange.util.ResultsWriter.append("your message")</code> .
PL-30566	The multi-threaded inbound integration <code>inbound-integration-config.xml</code> file now supports configurations that vary based on server environment set with the <code>env</code> system configuration setting. See the <i>Integration Guide</i> in the "Inbound Integration" chapter.
PL-30910	For custom implementations of the <code>InboundIntegrationStartablePlugin</code> plugin interface, all method signatures of the start and stop methods must call methods of <code>gw.api.integration.inbound.CustomWorkAgent</code> . As a method argument, pass the plugin name that you used in your <code>inbound-integration-config.xml</code> file with the <code>name</code> attribute on the <code><custom-integration></code> element. In each start method, call <code>CustomWorkAgent.startCustomWorkAgent(pluginName)</code> . In each stop method, call <code>CustomWorkAgent.stopCustomWorkAgent(pluginName)</code> . This new API does not apply to writing handlers for the built-in file or JMS integrations.
Other - Persistence Pod	
PL-28950	This release improves the performance of the <code>Populate Search Columns</code> batch process.
PL-30644	Added configuration parameter <code>BatchProcessPopulateSearchColumnsParallelDML</code> to enable or disable parallelism in the populate search columns batch process. By default parallelism is enabled.
Other - Web UI Pod	
PL-29767	There was an issue with localized date formats that used a single-digit day or month, which sometimes resulted in swapping the day and month. Single digits are now supported for these values. In addition, the following date separators are supported by default: <ul style="list-style-type: none"> / (slash; example: 18/2/2015) . (dot; example: 18.2.2015) - (dash; example: 18-2-2015) <space> (example: 18 02 2015) <nospace> (example: 18022015)
PL-31050	Fixed an issue where the <code>DateTime</code> input widget was not navigable with the keyboard.
Queries	
PL-29695	Fixed the <code>forEmpty</code> method on <code>Query</code> objects to work properly on root queries and have no effect on sub-queries.

ID	Description
Web - Configuration	
PL-29869	Fixed an issue where, in the RowTree widget, a Cell with a currency format type or a CurrencyCell would not show the currency format properly.
PL-30310	Fixed an issue where validation on a PrivacyInput was applying to the mask value instead of the actual value.
PL-30475	Fixed an issue where the LabelWidth property in an InputColumn was not being applied for specific widgets.
Web - IE Support	
PL-29681	Fixed an error where the server would report that the file toolbar-top-info-bg.gif was missing.
Web - ListViews	
PL-26386	Fixed an issue when postOnChange is enabled where tabbing through menu items or inputs with a menu icon would not show focus on those fields correctly.
PL-29294	Fixed an issue with a newly added list view column appearing at the extreme right until the layout preferences are reset.
PL-29657	Improved the behavior of tabbing through list view cells.
PL-29769	Fixed an issue where header menus were appearing incorrectly for list views with the colspan attribute set.
PL-30595	Fixed an issue where the screen would reset its horizontal scroll position while editing a list view in Internet Explorer.
PL-30989	Fixed an issue where the helper icon in a cell would not be visible when the cell had focus.
PL-31015	Fixed an issue where a LinkCell was not accessible with the keyboard.
PL-31038	List views now reposition themselves into view horizontally if you use the keyboard to navigate to an element off the screen.
PL-31152	Fixed an issue where you could not use the keyboard to invoke actions on text in list view cells.
PL-31201	Fixed an issue where the PrivacyInput menu remained expanded when navigating the page using the keyboard.
PL-31298	Fixed an issue where you could not open the menus of a ButtonCell with a MenuItem by using the Alt+Down Arrow keyboard shortcut.
PL-31354	Fixed an issue in list views where postOnChange would cause the focus to be set incorrectly.
PL-31380	Fixed an issue where focus would be lost if a postOnChange occurred in the last cell of a list view.
PL-31403	Fixed an issue where postOnChange was not triggered in list views with only one editable cell in Internet Explorer.
Web - Other	
PL-28929	Fixed an error message that referenced a missing "perf-analyzer". The analyzer is an internal Guidewire tool that is not used in customer releases.
PL-30507	Fixed an issue in the RunBatchProcessCommand to display an error instead of a crash if a work queue contains zero workers.
PL-31341	Fixed an issue where the no-store value was omitted from the Cache-control HTTP response header in application responses.
PL-31813	Fixed an issue where calling an ExitPoint would fail.
Web - Styling	
PL-28575	The workspace panel had a property that enabled it to float while hovering. This behavior led to a confusing user experience because the panel would appear to pop up and down on user actions. The hover behavior has been disabled, and a larger button for expand/collapse has been added.
PL-30112	Fixed an issue where the PreFormattedTextInput PCF widget did not properly render HTML.
PL-30534	Fixed an issue where headings in the Actions menu were not appearing in a bold font.
Web - UI/Runtime	
PL-28186	Fixed an issue where the PrivacyInput field would be exposed if used during postOnChange.

ID	Description
PL-28224	Fixed an issue where the PrivacyInput widget was not masking values that were changed by client-side reflection.
PL-28581	Fixed an issue where actions in the TitleBar were not displayed properly.
PL-28784	Fixed an error that occurred in scatter charts after data axes were hidden.
PL-29118	To prevent CSRF (Cross-Site Request Forgery) attacks BillingCenter now generates a unique Cross-Site Request Forgery (CSRF) token for each user session. The CSRF token is included in each request and used by the server to verify the legitimacy of the user request. See "Configuring Single Sign-on Authentication" in the <i>Installation Guide</i> .
PL-29230	Fixed a localization issue where the shadow placeholder text in the QuickJump box did not immediately refresh after the language was changed.
PL-29650	Fixed RangeValueInput to handle values that require HTML encoding (for example, an ampersand).
PL-29653	Fixed an issue where the screen would freeze or become blank when using typeahead in an AddressAutoFillInput widget.
PL-29689	Fixed an issue where detail views would lose focus after a postOnChange.
PL-29805	Fixed an issue where pressing Alt+Z to log out would do so even if there was unsaved work.
PL-29858, PL-30039	Fixed an issue with list views unexpectedly scrolling to the top after interacting with a radio button, check box, or combo box in a row.
PL-30055	Addressed an issue where keyboard input entered while a postOnChange event was in progress disappeared after the event completed. Keyboard input is now disabled until the event is complete.
PL-30072	Fixed an issue where postOnChange in the date widget required an additional user action before it would be triggered.
PL-30160	Fixed an issue where opening and then closing the DateTime widget would reset the minute value to the current minute of the user's machine.
PL-30207	Fixed an issue where you could not navigate out of a list view by using the Tab key.
PL-30492	Fixed an issue where using the Unsaved Work menu failed when a page title contained certain special characters (such as an ampersand).
PL-30586	Pressing the Backspace key no longer acts as a Back navigation command in the web browser.
PL-30772	Fixed an issue where the focus was not being set into a list view after a row was added.
PL-31001	Fixed an issue where the focus in a list view did not go to a newly added row.
PL-31027	Fixed an issue where list view rows with colspans specified did not handle tabbing correctly.
PL-31040	Fixed an issue where list view rows were highlighted incorrectly after columns were rearranged.
PL-31042	Fixed an issue where a list detail view configured to have an editable list view would create conflicting edits that would override each other and cause the data to be discarded. An impact of this change is that you now single-click a row to select it and double-click to edit it.
PL-31512	Fixed an issue where the calendar widget did not properly display Russian months.
PL-31610	Fixed an issue where reflection on the PCF widgets RangeRadioInput, BooleanRadioInput, and TypekeyRadioInput were not working properly.
Web Services - WSI (New)	
PL-31055	Fixed an issue in WSDL generation for services with multiple @Throws annotations for the same exception but different reasons. This issue caused an exception in some cases for WSDL consumers. The WSDL now merges related faults into one fault with all explanations concatenated in one comment. The fix prevents an exception that some WSDL consumers experienced.
Work Queues	
PL-31032	Improved performance of the work queue framework in finding available work items for work workers based on work item priority. The improvement introduces a new parameter, WorkItemPriorityMultiplierSecs, in config.xml, with a default value of 600 seconds.
PL-31043	Added BulkInsertWorkQueueBase for work queues with writers that typically select large volumes of work items. Instead of developing the work queue writer by implementing the findTargets method, implement the buildBulkInsertSelect method.
XMLElement (and XSD types)	

ID	Description
PL-28345	XSD types based on XMLNode have been deprecated, and will be removed in a future release.
PL-30793	Fixed an issue with WSDL import relating to handling simple data if a complexType is a complexContent extension of a complexType with a simpleContent extension.

Documentation Improvements and Resolved Issues

ID	Description
Batch processing	
DOC-2765	Added documentation on developing custom work queues. See "Custom Batch Processing" in the <i>Integration Guide</i> .
General	
DOC-8120	Rewrote and expanded the documentation describing the Charge Invoicing process and the ChargeInitializer plugin. Refer to the <i>Configuration Guide</i> chapter "Configuring the Charge Invoicing Process" and the <i>Integration Guide</i> section "ChargeInitializer Plugin."
DOC-10112	Rewrote and expanded the documentation describing Billing Levels and the related topics of Default and Designated Unapplied T-accounts. Refer to the new <i>Application Guide</i> chapter "Billing Levels."
DOC-10299	Accounts, Policies, and Producers are three integral parts of BillingCenter. Accordingly, each of these elements now commands its own chapter near the beginning of the BillingCenter <i>Application Guide</i> . Information about each element that was scattered throughout the documentation is now focused and consolidated in the new chapters.
DOC-10316	Rewrote and expanded the documentation describing Payment Allocation Plans. Refer to the <i>Application Guide</i> chapter "Payment Allocation Plans" and the <i>Configuration Guide</i> chapter "Configuring Payment Allocation Plans."
DOC-10317	Rewrote and expanded the documentation describing Suspense Payments. Refer to the <i>Application Guide</i> section "Suspense Payments and Items". Also, the <i>Integration Guide</i> section "Suspense Payment Plugin." Also, the <i>System Administration Guide</i> section "Suspense Payment."
Globalization	
DOC-7022	Concepts and configuration of address auto-fill and autocomplete are now described in the topic "Address Autocompletion and Autofill" in the <i>Globalization Guide</i> .
DOC-8027	Added a topic that has steps for adding a new address field. This example covers the classes, configuration files, and PCF fields that are involved in adding a new address field. Additionally, the example shows how to configure autofill and autocomplete functionality for the new field. See "Example: Adding a Country with a New Address Field" in the <i>Globalization Guide</i> .
DOC-8646	Added documentation describing how to localize Contact name information to "Configuring Name Information" in the <i>Globalization Guide</i> .
DOC-8869	Added documentation for how to edit, configure, and load zone data. See "Configuring Zone Information" in the <i>Globalization Guide</i> .
PCF / UI	
DOC-10093	New documentation on adjusting several aspects of the screen layout—such as the size, ordering, and visibility of list view columns—and managing these layout preferences. See: "Changing the Screen Layout" in the <i>Application Guide</i> .
Studio	
DOC-8672	Enhanced documentation on debugging the application within Studio. See "Debugging and Testing Your Gosu Code" in the <i>Configuration Guide</i> .
DOC-8873	Added documentation on viewing entity property values during debugging. See "Debugging and Testing Your Gosu Code" in the <i>Configuration Guide</i> .

Known Issues and Limitations

This topic describes known issues with this release of Guidewire BillingCenter.

This topic includes:

- “BillingCenter Known Issues” on page 154
- “Studio/Platform Known Issues” on page 158

Note: For maintenance releases, Guidewire often defers fixing configuration issues that require merging files during the upgrade. Workarounds for many of these issues are listed in the following topics. The goal of this policy is to make upgrades as straightforward as possible.

BillingCenter Known Issues

Rewrite Billing Instruction with the same policy number as the original policy is not incrementing the TermNumber and throws DBDuplicateKeyException (BC-2961)

Issue – If term number "1" already exists for the policy, sending a Rewrite billing instruction without specifying the term number will throw a `DBDuplicateKeyException` because BillingCenter uses the number "1" for term number.

Workaround – Include a term number that is unique for the given policy number when sending a rewrite billing instruction.

Total Payment Applied on agency bill statement wrong after payment plan edit (BC-2966)

Issue – The **Total Payment Applied** amount will be incorrect on the next agency bill statement if the following are also true:

- Producer has at least two policies
- Previous agency bill statement is billed and partially paid
- Payment plan used by one of the policies is modified

Workaround – Guidewire is aware of this issue.

Reversing write-off commissions and write-off commission recovery not in the user interface (BC-7184)

Issue – The BillingCenter user interface does not provide a way to reverse write-off commissions or write-off commission recovery.

Workaround – Clone the Reverse button under the Payments tab and reverse `CmsnPayableReduction` and other transactions using the following action: `CurrentLocation.startEditing()`, `producerTransaction.reverse()`, and `CurrentLocation.commit()`.

Commission: Commission override after a commission write-off causes write-off to be reserved (BC-9536)

Issue – Several methods of the `BillingAPI WS-I` webservice, such as `issuePolicyPeriod` and `cancelPolicyperiod`, indirectly call `BillingInstructionUtil.executeAndCommit`, which commits the current bundle. These methods also directly call `tryCatch`, which commits a new bundle after executing the entire block of code inside `tryCatch`. As a result, the code tries to commit the same bundle twice.

Following are two ways you can encounter this problem:

Case 1

If you try to create an activity in the block of code passed to `tryCatch` after the call to execute the billing instruction, BillingCenter throws `DBA1readyExecutedException`.

For example:

```
function issuePolicyPeriod(issuePolicyInfo : IssuePolicyInfo, transactionId : String) : String{
    require(issuePolicyInfo, "issuePolicyInfo")
    var publicID = tryCatch( \ bundle -> {
        var pid= issuePolicyInfo.executeIssuanceBI()
        // code *here* to create a new activity
        return pid
    }, transactionId)
    return publicID
}
```

Case 2

If you used `CreateTroubleTicketHelper` to create a trouble ticket in a billing instruction preupdate rule, you would get a `DBA1readyExecutedException`. The following is an example of a billing instruction preupdate rule that would trigger the error:

```
// create trouble ticket with a delq hold until second invoice
var troubleTicketHelper = new CreateTroubleTicketHelper()
var ticket = troubleTicketHelper.createTroubleTicket(
    billingInstruction as PlcyBillingInstruction).PolicyPeriod)
ticket.Title = "Delinquency hold on issuance"
ticket.DetailedDescription = ticket.Title
ticket.Priority = "normal"
ticket.TicketType = "automatic"
ticket.Hold.setAppliedToHoldType(HoldType.TC_DELINQUENCY, true)
ticket.Hold.setReleaseDate(HoldType.TC_DELINQUENCY, gw.util.GWBaseDateEnhancement.CurrentDate)
```

Workarounds

- For Case 1, in the block of code passed to `tryCatch`, do not create or update entities after the call to execute the billing instruction. For example:

```
function issuePolicyPeriod(issuePolicyInfo : IssuePolicyInfo, transactionId : String) : String{
    require(issuePolicyInfo, "issuePolicyInfo")
    var publicID = tryCatch( \ bundle -> {
        // code *here* to create a new activity
        var pid= issuePolicyInfo.executeIssuanceBI()
        return pid
    }, transactionId)
    return publicID
}
```

- For Case 2, instead of using `CreateTroubleTicketHelper` to create the trouble ticket, create related entities directly as in the following example in a PolicyPeriod enhancement:

```
var ticket = new TroubleTicket(this)
var ttJoinEntity = new TroubleTicketJoinEntity(ticket)
ticket.Title = ticketTitle
ticket.DetailedDescription = "Delinquency hold placed on Account with a " + ticket.Title
ticket.Priority = "normal"
ticket.TicketType = ticketType
ticket.Hold.setAppliedToHoldType( HoldType.TC_DELINQUENCY, true )
ticket.Hold.setReleaseDate(HoldType.TC_DELINQUENCY, releaseDate)
ticket.addToTroubleTicketJoinEntities(ttJoinEntity)
this.addToTroubleTicketJoinEntities(ttJoinEntity)
this.Account.addToTroubleTicketJoinEntities(ttJoinEntity)
ticket.Hold.checkForHoldAdditions()
```

PolicyCenter integration on renewal offer: suspense payment is not being applied to policy period (BC-9691)

Issue – When BillingCenter is integrated with PolicyCenter, a suspense payment that is created with an offer number from a PolicyCenter renewal is incorrectly applied to the owner account of the policy period. Correct behavior would be to apply the suspense payment to the renewal policy period instead of the owner account.

Workaround – Guidewire is aware of this issue.

Payment Distribution: BillingCenter incorrectly distributes payment amounts that have been disbursed (BC-9783)

Issue – Reversing a DirectBillPayment is causing T-account unapplied to become negative. This can happen when another direct bill payment on the same account gets modified.

Workaround – Create a recapture charge when unapplied goes below zero.

Net Owed Amount calculation for promises is sometimes incorrect (BC-9920)

Issue – The Net Owed amount is calculated incorrectly for invoice items that have been fully or partially promised.

Workaround – Guidewire is aware of this issue.

Saved promises are listed in two places after upgrade (BC-10019)

Issue – After upgrade, saved promises are also listed under Executed Promises location group.

Workaround – Execute saved promises prior to upgrade.

Inconsistency in behavior when an invoice item (with item level commission rate override) is removed from a charge (BC-10368)

Issue – When an invoice item with item-level commission rate override is modified, BillingCenter recalculates commission inconsistently depending on how the invoice item is changed.

For example, you can modify an existing invoice item on a charge. Alternatively, you can add a new invoice item to the charge and delete the existing invoice item. BillingCenter will calculate the resulting commission differently for these two cases.

Workaround – Guidewire is aware of this issue.

AccountImpl.getNotFullyAllottedFundsTrackers is not bundle safe (BC-11558)

Issue – Manipulating bundles for funds tracking should be avoided. All linked entities are not automatically removed when you remove entities from the bundle.

Workaround – Guidewire is aware of this issue.

Payment redistribution behavior in payment plan refactor changed (BC-12416)

Issue – In BillingCenter 7, payment plan refactor redistributed the entire payment. This behavior has changed as follows:

- In BillingCenter 8, payment plan refactor redistributes only the amount of the payment that had been originally allocated.
- In BillingCenter 7, payment plan refactor respects the account's distribution method. In BillingCenter 8, it does not because an account has no distribution method.

Workaround – If you would like to maintain BillingCenter 7 behavior, contact Guidewire BillingCenter Product Management.

Upgrading from BillingCenter 3.0.5 or earlier to 8.0 can result in policy commission errors (BC-13101)

Issue – In BillingCenter 3.0.5 and earlier, transferring a policy from one producer to a different one and then back to the original corrupted the data by creating duplicate policy commissions. Part of the fix involved an upgrade trigger that consolidated these policy commissions, but this upgrade did not redistribute the commission correctly to the individual invoice items afterward.

Workaround – If you are upgrading from BillingCenter 3.0.5 or earlier, before upgrading to BillingCenter 8.0, first do the minor upgrade to BillingCenter 3.0.6 or a later 3.0 release. After you do the upgrade to BillingCenter 8.0, contact Customer Support for SQL statements and Gosu code that fix any incorrect invoice items resulting from incorrectly redistributed commissions.

BillingAPI previewInstallmentsPlanInvoices method shows installments as deposits (BC-13639)

Issue – The BillingAPI web service method `previewInstallmentsPlanInvoices` is called by PolicyCenter when issuing a policy. When a Down Payment of 0% is specified for the policy, the installments are incorrectly shown as deposits.

Workaround – Make the following changes.

1. Update the `previewInstallmentsPlanInvoices` method in the `gw.webservice.policycenter.bc801.BillingAPI` web service to remove the `changeEarliestItemPreviewToDownPayment` method call.
2. Replace the `createInvoicesSummary` method in the `gw.webservice.policycenter.bc801.BillingInstructionInfoEnhancement` enhancement with the following:

```
function createInvoicesSummary(bi : PlcyBillingInstruction) : InvoiceItemPreview[] {  
    final var itemAmountsByPartitionMap = bi.PolicyPeriod.InvoiceItems  
        .partition(\ item ->  
            new gw.util.Pair<java.util.Date, InvoiceItemType>(item.InvoiceDueDate,  
                item.Type != TC_DEPOSIT and item.Type != TC_INSTALLMENT  
                    ? TC_INSTALLMENT : item.Type)  
)  
        .mapValues(\ partitionItems -> partitionItems.sum(bi.Currency, \ item -> item.Amount))  
  
    return itemAmountsByPartitionMap.Keys.map(\ partitionKey ->  
        new InvoiceItemPreview(null, partitionKey.First, null,  
            itemAmountsByPartitionMap[partitionKey], partitionKey.Second)  
    ).toTypedArray()  
}
```

Reversing a ProducerPayableTransferTxn may generate inaccurate producer statements (BC-13821)

Issue – When using Gosu to reverse a `ProducerPayableTransferTxn` entity, expected behavior is for the transaction to be placed on the next planned statement of each producer. However, depending on the timing of the reversal, the transaction for the receiving producer may be incorrectly placed on a statement that has already been sent. If this occurs, the Previous Balance of subsequent statements will contain inaccurate values.

To determine if your database has been affected by this issue, run the following query:

```
select count(*)  
from bc_producercontext ctx, bc_producerstatement stmt  
where ctx.PayableReceiverStatementID = stmt.ID  
and ctx.CreateTime > stmt.StatementDate
```

Workaround – If this query returns a non-zero count, contact Guidewire Customer Support for assistance.

New contact created in BillingCenter for account created in PolicyCenter causes exception (PC-23873)

Issue – When BillingCenter creates a new Contact for an Account that had been created in PolicyCenter, PolicyCenter does not recognize the Contact and throws a `BadIdentifierException`.

Workaround – Guidewire is aware of this issue.

Cancellation charges can create orphaned invoice entries (BC-14422)

Issue – When a pro-rata cancellation charge results in a refund and partial cancellation charge, orphaned invoice entries are created. If a subsequent change to the payment plan is attempted, the orphaned entries cause the following exception: `java.lang.IllegalStateException: Cannot revert an entry in the orphaned state.`

Workaround – Modify the `SpecialHandlingChargeSlicer.recreateInvoiceItems` method so it does not revert entries in an orphaned state. The complete workaround implementation is shown below.

```
// *** SpecialHandlingChargeSlicer.gs source file ***
override function recreateInvoiceItems( changer : ChargeInstallmentChanger,
                                         totalAmountForInvoiceItems : MonetaryAmount ){
    for( var entry in changer.Entries ){
        // Revert entry only if it's not in an orphaned state
        if( !entry.InOrphanedState ){
            entry.revert()
        }
    }
}
```

Invoices inaccurately processed following multiple changes to payment plan periodicity (BC-14513)

Issue – If a policy’s payment plan periodicity is changed multiple times (such as from monthly to weekly and then back to monthly), and the policy has a commission plan that earns commissions early (such as upon binding or first payment), the resulting invoice-recalculation process does not generate the expected installments. Instead, a single installment is generated for the total amount of all planned invoice items (excluding any down payment amounts) and the invoice is dated toward the end of the policy period.

Workaround – Modify the invoice items manually in the **Charges** screen by selecting the **Account** or **Policy** tab’s **Charges** → **Modify Invoice Items** button.

Studio/Platform Known Issues

Renaming method or property throws ParseResultsException (PL-16633)

Issue – If you rename a property or a method or you change a method signature, and a workflow references that property or method in a Gosu field, BillingCenter throws `ParseResultsException`. This is the intended behavior.

Workaround – Restart the workflow engine. To do so:

1. Log in to BillingCenter using an administrative account.
2. Access **Internal Tools** → **Reload**.
3. Click **Reload Workflow Engine**.

Long text in table cells can add white space to the right of the page (PL-28288)

Issue – In an editable list view, extremely long text entered in a single cell can cause additional white space on the right side of the page. Long text is text that occupies approximately the entire width of the screen. This issue occurs primarily in Chrome.

Workaround – If you expect users to enter large amounts of text into the cells of a column, configure the column to support text wrapping.

Administrative command-line tools cannot refresh WSDL (PL-29021)

Issue – Administrative command-line tools rely on web service implementation classes such as `MaintenanceToolsAPI.gs`. If you change the web service implementation classes, administrative tools might fail because the tool’s WSDL does not match the server WSDL. Some changes do not affect the WSDL. For example, adding a `@WsPermission` annotation.

Workaround – Do the following:

From the `BillingCenter/bin` directory, at a command prompt type the command:

```
gwbc regen-soap-api
```

In Windows Explorer, copy the WSDL files from the location:

BillingCenter/soap-api/wsi/wsdl

to:

BillingCenter/admin/res/wsi

No entity changed event fired through a one-to-one entity relationship (PL-26224)

Issue – The existence of a one-to-one entity relationship between two entities prevents an entity change event from being fired, even if it is supposed to. For example, in ContactManager there is a foreign key from ABContact to Address for the primary address, and a one-to-one relationship from Address to ABContact. Because of the one-to-one link, no event changed event is fired for any change in the primary address.

More generally, if entity E1 has a one-to-one relationship to entity E2, no entity changed event is fired for E2 if a property on E1 changes.

Workaround – You might be able to take advantage of a secondary change caused by the initial change, with the secondary change triggering the event to be fired instead. In the previous ContactManager example, making sure that changes to all properties in Address cause a History record to be generated ensures that the entity changed event will be fired.

Chrome browser cannot display product documentation in HTML format (DOC-7251)

Issue – If you use the Google Chrome browser, you can view the HTML Guidewire product documentation only if it is served by an HTTP server using the `http://` protocol. The Chrome browser cannot load HTML product documentation from your local disk by using the `file://` protocol.

Workaround – Use a different browser, such as Microsoft Internet Explorer. If you use the HTML documentation only to access the PDF files, you can open them directly in the `pdf` subdirectory of the `doc` directory.

New mechanism for reloading Gosu classes (DOC-8218)

Issue – In past releases, you could modify your PCF files and Gosu classes in Studio, and then reload the changes into the running server by pressing `Alt+Shift+L` in the application user interface. This shortcut no longer loads Gosu classes.

Workaround – To have the server reload your Gosu classes, in Studio, click **Build** → **Make Project**. When Studio is finished compiling your project, the changes will be loaded. You can also restart your server to load the Gosu classes.

Alignment Property Does Not Work for Inputs in a DetailView (PL-29429)

Issue – Inputs in a detail view do not align to the right or center. They always align to the left, regardless of the value of the `align` property.

Workaround – If you need to right align a series of items such as a set of monetary values, put them into a list view instead of a detail view. You can then set the alignment as needed.

JMS inbound integration implementations must be OSGi or Java plugins (PL-32054)

Issue – Writing JMS inbound integration handlers requires casting an object to the type `javax.jms.Message`. In Gosu, for some app servers this fails with the message: "Caused by: gw.lang.parser.exceptions.ErrantGosuClassException: GosuClass test.GosuJMSMessageReplyHandler has errors, and cannot be used at runtime."

Workaround – Implement JMS handler code for inbound integration in Java. Guidewire recommends that for inbound integration Java code, implement your code as an OSGi plugin, not a standard (non-OSGi) Java plugin.



chapter 18

Guidewire BillingCenter 8.0.2 Release Notes

Release 8.0.2

Release Notes Update

IMPORTANT These release notes replace the release notes that were included with the official product release. Please disregard the earlier version of the release notes.

Document Revision: 02-July-2015

Copyright © 2001-2015 Guidewire Software, Inc. All rights reserved.

Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire Reinsurance Management, Guidewire ContactManager, Guidewire Vendor Data Management, Guidewire Client Data Management, Guidewire Rating Management, Guidewire InsuranceSuite, Guidewire ContactCenter, Guidewire Studio, Guidewire Product Designer, Guidewire Live, Guidewire DataHub, Guidewire InfoCenter, Guidewire Standard Reporting, Guidewire ExampleCenter, Guidewire Account Manager Portal, Guidewire Claim Portal, Guidewire Policyholder Portal, Gosu, Deliver Insurance Your Way, and the Guidewire logo are trademarks, service marks, or registered trademarks of Guidewire Software, Inc. in the United States and/or other countries.

All other trademarks are the property of their respective owners.

This material is confidential and proprietary to Guidewire and subject to the confidentiality terms in the applicable license agreement and/or separate nondisclosure agreement.

Guidewire products are protected by one or more United States patents.

Overview

These release notes contain the following topics:

- Release Information
- Installing This Release

- Support
- Major Issues and Changes
- Improvements and Resolved Issues
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire BillingCenter.

IMPORTANT If you skipped one or more upgrade releases to BillingCenter, be sure to read the Release Notes for those releases to learn about changes and fixed issues.

Version Number

This release of Guidewire BillingCenter is 8.0.2.

- If you are integrating with Guidewire PolicyCenter 7, this version of BillingCenter requires PolicyCenter 7.0.4 or later. The latest maintenance release is preferred.
- If you are integrating with Guidewire PolicyCenter 8, this version of BillingCenter requires PolicyCenter 8.0.2 or later. The latest maintenance release is preferred.
- If you are integrating with Guidewire ContactManager, this version of BillingCenter requires ContactManager 7.0.0 or later. The latest maintenance release is preferred.

Installing This Release

Refer to the following:

- The *BillingCenter Installation Guide* for general installation information
- Prior BillingCenter release notes for any versions that you have skipped

Upgrade Information

Upgrading from BillingCenter 3.0.4 or newer versions is supported in this release.

This topic contains the following:

- Incorrect warning message when running Configuration Upgrade Tool (PL-28723)
- You can now nest block comments in Gosu code (PL-28663)

Incorrect warning message when running Configuration Upgrade Tool (PL-28723)

When upgrading from one BillingCenter 8 release to another BillingCenter 8 release, running the Configuration Upgrade Tool may produce the following warning message:

WARN cannot find Emerald base configuration zip, this could indicate a problem...

If the `BillingCenter/modules/base.zip` file does exist, then you can safely ignore this warning.

You can now nest block comments in Gosu code (PL-28663)

Block (multiline) comments in Gosu code can now be placed within other block comments. This is often useful for commenting out blocks of code that already have existing comments within them. For example:

```
/*  
if(x) {
```

```
    print("x is true") // success
} else {
/* failure */
    print("x is not true")
}
*/
```

The ability to nest block comments also allows you to comment out methods that contain Javadoc, which is written within comment blocks.

This change may have an unexpected impact on existing code that contains comment symbols within comments. In previous releases, those symbols were ignored. With this release, they may now be interpreted as new comments. For example, consider the following function:

```
1 function test(){
2     /*
3      */
4     */
5     print("hello")
6 }
```

In previous releases, line 3 was ignored because it appears within a comment block. It is now interpreted as the opening of an additional comment block, nested within the outer comment. However, because the nested comment block has no close, the above code now results in an error such as “unclosed comment”.

As another example, lines such as the following may be used to visually separate methods or sections of code:

```
1 /*
2 //*****
3 */
```

In previous releases, line 2 was ignored because it appears within a comment block. Now the character sequence /* in line 2 is interpreted as the opening of an additional comment block, one that has no close. The single-line comment sequence // is ignored within a block comment.

To avoid this issue, make sure that the body of a code comment does not contain the character sequence /* unless there is also a corresponding */ character sequence.

Support

For assistance, visit the Guidewire Resource Portal – <http://guidewire.custhelp.com>

Major Issues and Changes

This topic contains major issues and changes that can affect your installation. For information on new features and other changes, see the topic “New and Changed in BillingCenter 8.0.2” in the BillingCenter New and Changed Guide.

This topic includes:

- Base PCF File Changes
- Changes in This Release Provided in Upgrade Diff Report

Base PCF File Changes

The following link requires that the `ReleaseNotes_files` directory be on your local disk in the same directory as this release notes file.

BillingCenter release 8.0.0 to 8.0.2

To view a report of the changes to the base PCF files, *[click here](#)*.

Changes in This Release Provided in Upgrade Diff Report

Guidewire provides a report detailing certain differences between the current release and your prior release. This report describes changes in display keys, entities, typelists, and the Gosu API. To obtain your custom Upgrade Diff Report, visit the Guidewire Resource Portal at <http://guidewire.custhelp.com>.

Improvements and Resolved Issues

BillingCenter Improvements and Resolved Issues

This topic describes improvements and issues corrected in this release. Guidewire attempts to provide information for issues of primary importance to our customers. This list is not intended to be comprehensive.

ID	Description
Account	
BC-3968	During an edit session of the New Account page, if you navigate to another page without creating the account, the edit-session data is saved to the Unsaved Work queue. In previous versions, the data could be lost unless Autosave was active.
BC-13346	The Account → Summary would display inaccurate Past Due notifications if the current date matched the DueDate of the first invoice. This issue has been repaired.
Accounting Subsystems	
BC-13253	<p>Added the configuration parameter EnableChargeProRataTxCreation to disable creation of ChargeProRataTx entities. BillingCenter normally creates ChargeProRataTx entities when a Charge is created with a ChargePattern subtype of ProRataCharge. These entities are subsequently processed by the ChargeProRataTx batch process to track earned premium. If you do not use BillingCenter to track earned premium, the ChargeProRataTx entities are not needed. When EnableChargeProRataTxCreation is set to false, ChargeProRataTx entities are not created.</p> <p>In BillingCenter 8, the default value of EnableChargeProRataTxCreation is true. If EnableChargeProRataTxCreation is set to false, it cannot be enabled again at a later time.</p> <p>Note: The EnableChargeProRataTxCreation parameter and ChargeProRataTx batch process are used only to track earned premium. If you do not use BillingCenter to track earned premium, set EnableChargeProRataTxCreation to false, and do not run the ChargeProRataTx batch process. Doing so reduces data storage usage and improves system performance. Any existing ChargeProRataTx entities not processed by the ChargeProRataTx batch process are ignored by BillingCenter.</p>
Batch Processes	
BC-9322	The following batch processes were unused and have been removed from the system: <ul style="list-style-type: none"> - Bulk Purge - Solr Data Import - Stat Report - Statistics
Credit Distribution	
BC-13655	<p>Credit distributions can now be reversed. Reversed credit distributions cannot be redistributed. Instead, create a new credit distribution to allocate them.</p> <p>Also, credit distributions can no longer be undistributed. Instead, reverse the credit distribution.</p>
Delinquency / Delinquency Plan	
BC-13286	Repairs an issue relevant to BillingCenter 8.0.1 only: When a payment for a past-due item is reversed, a delinquency on the policy period did not start.
BC-13414	<p>On the Desktop → My Delinquencies page, the default filter is changed from All open to New opened (this week). The former default filter could potentially return a large number of results, negatively affecting performance.</p> <p>Also on the My Delinquencies page, the filters All open owned and Closed in last 30 days have been removed and deprecated. In the base configuration, these filters never returned any results. If these filters have been configured in your system to return results, contact Guidewire Customer Support for assistance.</p>
BC-13438	<p>The Desktop → My Delinquencies page sorts delinquencies by start date with the newest entry first. Previous releases showed delinquencies in no specific order.</p> <p>The New opened (this week) filter was modified to be consistent with the All open filter. Both filters search for DelinquencyProcess objects having a Status field setting of Open, Error, or OnHold.</p>
Disbursement	

BC-13561	Repairs an issue relevant to BillingCenter 8.0.0 and 8.0.1 only: If an Account has multiple Unapplied T-accounts, the AutomaticDisbursement operation could create multiple work items for the Account, rather than the desired single work item.
General	
BC-11671	When searching for a document to attach to an email, the operation succeeds provided the IDocumentMetadataSource plugin is enabled. By default, however, IDocumentMetadataSource is disabled. When disabled, the document search operation throws an UnsupportedOperationException. This is a known issue.
BC-13473	The following entity enhancements are writable and available for configuration, as they are in BillingCenter 8.0.0 and earlier releases. They were temporarily read-only in BillingCenter 8.0.1. <ul style="list-style-type: none"> - ActivityExt.gsx - BCBigDecimalExt.gsx - BCDateExt.gsx - BCMonetaryAmountEnhancement.gsx - BCStringExt.gsx - DelinquencyProcessExt.gsx - DelinquencyTargetExt.gsx - InvoiceItemEnhancement.gsx - InvoiceItemTransactionExt.gsx - NewMultiPaymentExt.gsx - PaymentPolicyExt.gsx - PolicyPeriodExt.gsx - PremiumIncentiveExt.gsx - ProducerExt.gsx - StatementInvoiceExt.gsx
Integration	
BC-12265	When creating a new Producer with the New Producer Wizard, added Contacts are accurately shown. In previous versions, some Contacts were occasionally not shown until the page was refreshed.
BC-13209	When calling the BillingAPI.issuePolicyPeriod method, the issuePolicyInfo argument must specify an existing ProducerCodeOfRecordId. If the producer does not exist, the method throws a BadIdentifierException. In previous versions, the policy was issued without a producer.
BC-13241	Added a BillingAPI web service method, getProducerInfo. The getProducerInfo method accepts a ProducerID argument and returns a ProducerInfo object. The ProducerInfo object includes a new field called AgencyPlanInfos which contains a list of AgencyBillPlanInfo objects associated with the ProducerID.
BC-13243	Added a BillingAPI web service method, getProducerCodeInfo. The getProducerCodeInfo method accepts a ProducerCodeID argument and returns a ProducerCodeInfo object. The ProducerCodeInfo object has a new field called CommissionPlanInfos which contains a list of CommissionPlanInfo objects associated with the ProducerCodeID.
BC-13296	The method ContactEnhancement.syncWithContactManager now returns the locally existing version of the contact if the contact could not be found in ContactManager. Previously ContactEnhancement.syncWithContactManager returned null if the contact was not found.
BC-13332	The BillingAPI methods createProducer and updateProducer will ignore any duplicate Agency Bill Plan or Commission Plan IDs. In previous releases, a BadIdentifierException was thrown.
BC-13355	The BillingAPI webservice method getUnappliedFunds supports multiple currencies. Use the getUnappliedFunds method to get all Unapplied funds, regardless of currency. The BillingAPI webservice method getUnapplieds supports only a single currency, and you can continue to use it for single currency implementations, or in multicurrency implementations you can use getUnapplieds to get only the Unapplied funds that match the currency of the account.
BC-13402	The BillingAPI methods renewPolicyPeriod and rewritePolicyPeriod correctly create a new policy period under all conditions. In previous versions, removal of an overriding payer or overriding invoice stream before renewing or rewriting could prevent successful policy period creation.
BC-13416	If the BillingAPI.getAllSubAccounts method specifies an accountNumber that does not exist, the method returns an empty array. In previous versions, the method threw a BadIdentifierException.
BC-13472	When creating an account in BillingCenter or PolicyCenter, if a Billing Method is not specified, a default value of Policy Level Billing with Designated Unapplied is used.
BC-13483	Added the PolicyTermInfos property to the PolicyBillingSummary DTO object returned by the BillingSummaryAPI method retrievePolicyBillingSummary. The PolicyTermInfos property contains an array of PolicyTermInfo DTO objects. Each PolicyTermInfo object includes properties that identify the billing period.

BC-13486	The following entities were returning a localized string instead of a global code for the Currency property. This problem has been fixed. <ul style="list-style-type: none"> - PlanCurrencyInfo - AgencyBillPlanInfo - CommissionPlanInfo - PaymentPlanInfo
BC-13503	Updated the BillingAPI methods getAllPaymentPlans and getPaymentPlansForAccount. The returned PaymentPlanInfo.AllowedPaymentMethods includes only payment methods relevant for recurring payments, such as ACH, CreditCard, and Responsive.
BC-13528	The BillingSummaryAPI.retrievePolicyBillingSummary method returns a PolicyBillingSummary object that includes all policy periods, regardless of the currency of each policy period. Previously, the returned PolicyBillingSummary object included only policy periods with currency that matched the currency of the policy period passed to the method.
BC-13535	If the ContactAPI.updateContact method is called with an AddressBookUID that is not in the BillingCenter database, BillingCenter now correctly ignores the update request. In BillingCenter 8.0.1, the same situation threw a BadIdentifierException.
BC-13661	This issue affects only users of multicurrency in BillingCenter 8.0.1. When changing the currency during the renewal or rewriting of a policy that includes an overriding InvoiceStream, but without an overriding Payer, BillingCenter 8.0.1 threw an IllegalArgumentException.
BC-13674	When processing an invoice-preview operation, BillingCenter will accept an unknown Billing Account that has not been created yet. In previous versions, a BadIdentifierException would be thrown.
BC-13677	When creating a Producer using the BillingCenter API, the PrimaryContact field in the PCNewProducerInfo object is optional. In previous versions, BillingCenter required the PrimaryContact field and would throw a NullPointerException if it was not specified.
BC-13723	Corrected the order of operations during the List Bill and Direct Bill renewal and rewrite processes, thereby preventing an IllegalStateException being thrown.

Internationalization

BC-9023	On the Desktop → My Activities page, the right-most drop-down list filter's Attempted Writeoff text can be localized. In previous versions, the text was hard-coded.
BC-9527	On the Administration → Monitoring → Message Queues page, the Destination text strings PAS, Document Store, and Contact Message Transport can be localized. In previous versions, these strings were hard-coded.

Payment

BC-2722	For each account, the NewPayment batch process bundles all received payments and creates a single work item to handle the resulting collection. Grouping in this manner reduces the possibility of encountering a ConcurrentDataChangeException situation. Previous versions created a separate work item for each payment received.
BC-13504	Removed the payment methods Wire and Misc from available options for use in creating a new payment instrument for an account or an invoice stream. These payment methods remain available if you are creating a new payment instrument for a direct bill payment.

Payment Distribution

BC-12470	Subtypes of BaseMoneyReceived must be distributed before being executed. The only exceptions to this are AgencyBillMoneyRcvd and DirectBillMoneyRcvd. Also, subtypes of BaseMoneyReceived must be executed before being committed. The only exceptions to this are AgencyBillMoneyRcvd, its subtypes, and PromisedMoney which can all be committed without being executed, provided they have been distributed.
BC-13477	You can now modify the distribution of a payment by a list bill account. Previously, you could not modify a list bill payment because BillingCenter failed to display the invoice items that were originally paid.

Payment Plan and Invoicing

BC-9595	During a payment plan change on an agency bill policy, if an existing payment distribution overpaid commission, the overpayment of commission is now placed on the first recalculated invoice item. The producer unapplied balance is not changed. Previously, the excess commission was incorrectly deducted from producer unapplied.
---------	--

BC-13141	After switching an account's billing level, the generated invoice numbers are assigned in ascending order based on the date. In previous versions, invoice numbers were assigned in the order the invoices were created within BillingCenter.
BC-13303	The ChargeInstallmentChanger class has a new field NewPaymentPlan, which gives access to the new payment plan in the IPaymentPlan.recreateInvoiceItemsForPaymentPlanChange plugin method during a payment plan reorganization.
BC-13343	After modifying a Payment Plan, the installment numbers associated with Invoice Items are renumbered to accurately reflect the payment change.
Performance	
BC-13378	<p>BillingCenter now uses the MaxSearchResults configuration parameter.</p> <p>Search results can be limited to a maximum quantity by setting the MaxSearchResults configuration parameter. Searches that match more than MaxSearchResults display a message asking the user to refine the search criteria.</p> <p>The MaxSearchResults parameter is defined in config.xml. Its default value is 300. Setting the value to -1 disables its effect for all search screens. An individual search screen can define a custom value by editing the maxSearchResults value in its PCF file.</p> <p>The MaxSearchResults parameter has no effect in two BillingCenter search screens. The Contact screen recognizes its own configuration parameter called MaxContactSearchResults. The Payment screen includes multiple result tabs which does not lend itself to a single MaxResults-type parameter.</p>
BC-13698	<p>Significant performance improvement achieved when running the New Payment and Automatic Disbursement batch processes.</p> <p>If upgrading from BillingCenter 8.0.0 or 8.0.1 and you have a large database, the upgrade may take longer than normal for a maintenance release. This is because new indexes and histograms are being created for several tables, specifically bc_account, bc_taccount, and basemoneyreceived.</p>
Producer	
BC-13011	Deprecated the Producer fields TotalOutstandingAmountOnAccounts and TotalOutstandingAmountOnPolicyPeriods, which were being used internally by Guidewire only. The recommended replacement is the Producer field OutstandingAmount.
SOAP APIs	
BC-13514	Reinstated the TroubleTicketAPI.putHoldOnPolicy method. The method was temporarily deprecated in BillingCenter 8.0.1.
Staging Table	
BC-13258	In order to improve staging table performance, a change was made to require the Item Event staging table bcst_itemevent to specify a setting for the Currency field.
BC-13290	Fixed a problem where the InitialChargeTxn transaction entity created for collateral requirement charges was missing the line item pointing at the Collateral Reserve T-Account.
BC-13302	This change improves the performance of loading InvoiceItem data from staging tables. BillingCenter now uses an intermediate temporary table to construct many of the denormalized values maintained on InvoiceItem and then joins with this table when inserting data into the main tables. The only visible manifestation of this change to you is that it will cause a database upgrade, because columns that used to be maintained in the staging tables as overwritten in staging columns have now been moved into the temporary table.

Platform Improvements and Resolved Issues

Activities, Assignment	
PL-29201	A new permission has been added to allow users to assign activities from a queue. This permission is not assigned to any roles by default. To allow users to assign activities from a queue, add the new permission actqueueassign or SystemPermissionType.TC_ACTQUEUEASSIGN to the necessary role.
Archiving	

PL-28043	Fixed an archiving bug that caused improper handling of a data model change during restoration of an entity from the archive. If the data model was changed so that a property was moved from a subtype to its supertype, the related data on the property could have been lost from the archive document during restoration.
PL-28098	Improved archiving to support links to objects outside the current graph of archived objects. In your <code>IDataModelUpgrade</code> plugin implementation, every <code>IArchivedEntitySet</code> object has two new methods. The <code>addCrossGraphRelationship</code> method adds a cross-graph relationship between a <code>publicID</code> field and a link field. The <code>removeCrossGraphRelationship</code> method removes the relationship.
PL-29160	Improved archiving graph validation error message related to overlap tables to simplify error messages to only include entities that are the root of issues and also include the involved links in the error message.
Build Infrastructure	
PL-24111	When generating a WAR file using the build-war command and passing <code>-Dconfig.war.dictionary=true</code> , the data dictionary was not being generated. This issue is resolved.
Clustering	
PL-29135	<p>In the event of a serious network failure (such as a network switch failure) when a cluster is split into many parts, JGroups might have problems merging these parts into a single cluster after the network is restored. To solve this potential problem Guidewire implemented a JGroups watchdog timer that resets the JGroups channel if it thinks that the node is running outside of the cluster for a long time.</p> <p>The JGroups watchdog uses the following algorithm:</p> <ul style="list-style-type: none"> Coordinator of the cluster sends periodic heartbeat messages. Each message contains the list of the cluster members from the coordinator's point of view. The watchdog resets the channel if the node does not receive messages from its coordinator for a long time. Note that an incorrect heartbeat (one that does not include the node in the list of members) is equal to a missing heartbeat. <p>This feature can be configured by adjusting two new parameters in <code>config.xml</code>:</p> <ul style="list-style-type: none"> <code>JGroupsWatchdogHeartbeatIntervalSecs</code> - Coordinator node sends a heartbeat each interval. Other nodes reset JGroups channel if they do not receive the heartbeat from their coordinator for several intervals. The default value is 30 seconds. <code>JGroupsWatchdogMissedHeartbeatsBeforeReset</code> - The number of missed heartbeat messages after which a node resets its JGroups channel. The default is 10.
PL-29240	<p>Updated the default value for <code>ClusterProtocolStack</code> to a configuration that is more reliable in case of a major network failure.</p> <p>FD protocol was replaced with FD_ALL, MERGE2 was replaced with MERGE3.</p>
Configuration Upgrade	
PL-19432	The configuration upgrade tool now accepts Y or N keyboard input for dialog boxes.
PL-28639	The configuration upgrade tool now formats metadata extension files in a more consistent way.
Consistency Checker	
PL-26560	Added a progress bar to the consistency check page.
PL-26841	Fixed an issue where a check constraint error might not show the description of the check that failed.
PL-28361	If one or more consistency checks result in an SQL failure, then the run will be flagged to allow a rerun of just the checks that failed, in a single thread.
PL-28579	Consistency checks are now executed by the worker threads individually instead of being grouped by table and consistency check type.
PL-28819	If an application server crashes while consistency checks are being run, when the server restarts, the worker threads will restart, but some of the consistency checks won't be run.
Consistency Checker, Other - Database Pod	
PL-28612	Fixed an issue where a work queue could wait a long time if the server is restarted when an item is being processed.
Data Distribution	
PL-28620	Fixed an issue that would cause the data distribution process to fail.
Database Configuration, Database Instrumentation	
PL-28478	Database debug logging including SQL statements has been enabled for JNDI configurations.
Database Configuration, Database Support - General, Database Support - Oracle, Database Upgrade	

PL-28601	Resolved an issue in which the database statistics configuration in config.xml was not honored during upgrade. Statistics for tables that were specified to keep their statistics were being erroneously deleted.
----------	---

Database Configuration

PL-28660	The validationQuery attribute has been removed from the database configuration of the DBCP connection pool.
----------	---

Database Configuration, Database Support - Oracle

PL-28730	Updated the Database Parameters/Guidewire Database Config Statistics Settings page to show only the configured items.
----------	---

Database Instrumentation

PL-28236	The Server Tools → Database Statistics page has been simplified. In addition, the code that compared the actual row counts with the calculated row counts, and used that information to calculate the statistics, has been eliminated. Statistics can be run unconditionally on all or a set of tables, or you can run incremental statistics, which uses the incrementalUpdateThresholdPercent attribute of the databaseStatistics configuration element, which defaults to 10 percent, to decide which statistics to run. Now the database statistics report reports all the update statistics statements that would be executed.
PL-28294	Fixed an issue where system_tools and maintenance_tools could be run from the command line without proper permissions.

Database Support - General

PL-28655	Improved the performance of proximity searches.
----------	---

Database Support - Oracle

PL-29300	The enable-all setting of the ora-parallel-dml attribute on the upgrade element no longer forces parallel execution in Oracle for an InsertSelectBuilder that is used outside of upgrade. The BeforeUpgradeInsertSelectBuilder is still forced to use parallel execution if the ora-parallel-dml attribute is set to enable-all.
----------	--

Database Support - SQL Server, Database Upgrade

PL-28779	Fixed an issue that prevented upgrade to a BillingCenter 8.0 maintenance release when the database type is SQL Server.
----------	--

Database Support - SQL Server

PL-28885	Fixed an issue which allowed users to create the BillingCenter schema on a SQL Server system database.
----------	--

Database Upgrade

PL-28302	The Upgrade Info report is now generated even when the upgrade fails, if the upgrade is run in development mode. The report is generated in the default temporary-file directory specified by the system property java.io.tmpdir. On UNIX systems the default value of this property is typically /tmp or /var/tmp. On Microsoft Windows systems it is typically c:\temp.
PL-28447	If data model files were changed but the extension.properties version number was not updated, BillingCenter would report an error when starting the server. This error did not include the file name of the extension file that included the data model change. The error has been updated to include the extension file name.
PL-28463	Resolved an issue that caused a ClassCastException when trying to delete a column using the method dropColumns inside an AfterUpgradeVersionTrigger.
PL-28572	Improved the performance of the upgrader when typecodes have been removed.
PL-28613	Fixed the ordering of typelist IDs, which could create a race condition during schema verification while upgrading the database.
PL-28773	We now support database upgrade version trigger configuration options to cause select statements, particularly in VersionChecks, to use an Oracle hint to cause parallel execution in query (SELECT) statements.
PL-29200	If the upgrade configuration attribute deferCreateArchiveIndexes is not explicitly specified in the database-config.xml file, then its default value has been changed to false. See the <i>Upgrade Guide</i> topic "Deferring Creation of Archive Indexes" for information about this feature.

Entities/Metadata

PL-29932	Added the following attributes to the MonetaryAmount entity: createhistogram, createAmountHistogram, createCurrencyHistogram.
----------	---

Gosu	
PL-28349	Previous versions of Gosu reported a warning on ambiguous method calls. Ambiguous method calls can hide a logical bug in your code. Previously, the Gosu compiler selected the best matching method to remove ambiguity. This release changes behavior in two ways (1) Ambiguous calls are now an error instead of a warning. (2) Studio now has a code inspection to identify and optionally fix any ambiguous code to previous Studio behavior. This inspection is disabled by default. To find and fix potential logical errors, Guidewire recommends that you run the inspection and carefully individually analyze every ambiguous call before applying any proposed fix.
PL-28663	You can now nest comments in Gosu code.
Gosu, IntelliJ IDE - Gosu Editor	
PL-28777	Fixed an issue where Studio would display invalid implicit conversion to integer warnings on PCF code that used Gosu reflection.
IntelliJ IDE - Customer Build	
PL-29861	The version number reported by Guidewire Studio now reflects the particular Studio version rather than the version of its companion Guidewire application. These numbers may be different.
IntelliJ IDE - Entity Editor	
PL-28783	Fixed an issue where changes to a delegate containing MonetaryAmount would not be saved correctly.
PL-29211	Fixed an error that would occur when trying to open several typelist editors at the same time.
IntelliJ IDE - Entity Editor, IntelliJ IDE - Typelist Editor	
PL-28999	Removed the file name suffix option from the entity extension and typelist extension dialog boxes, since this is not a common operation, and removing the option simplifies the creation process. To restore this option, click File → Settings, then navigate to the Guidewire Studio → Metadata Editor page, and then set Show filename suffix on new extension dialog.
IntelliJ IDE - Entity Editor, IntelliJ IDE - Typelist Editor	
PL-29592	Fixed an error that would occur if you tried to create an entity or typelist extension when one already exists.
IntelliJ IDE - Formatting	
PL-29310	When viewing PCFs, entities, and typelists as XML in Studio, unicode characters now display according to their language native symbol, rather than as a unicode escape sequence.
IntelliJ IDE - Other	
PL-27965	The Compare to Base and Show Diff windows can now be resized and maximized.
PL-29172	Guidewire Studio is bundled with IntelliJ IDEA Community Edition. If you would like to run Studio with your own installation of IntelliJ IDEA Ultimate Edition, you can now do so. Only version 12.1.7 of IntelliJ IDEA Ultimate Edition is supported. To enable this, in the Guidewire product root directory, create a text file named studio.ultimate that contains the path to the IntelliJ IDEA Ultimate installation directory.
IntelliJ IDE - Typelist Editor	
PL-28289	Fixed a problem when, after installing a language pack, typelists within the language pack were not recognized as valid metadata files.
Internal/Server Tools Pages, Work Queues	
PL-28725	Implemented the Work Queue management bean. The Management Beans page now shows a management bean for each work queue running locally.
Localization	
PL-27731	The maximum length of a phone number extension can now be set in the Plugin Registry editor. The default length is 4. To set this value:
	1. Open Guidewire Studio and navigate in the Project window to configuration → config → Plugins → Registry → iPhoneNormalizerPlugin.gwp.
	2. Add a new parameter by clicking the + button to the right of the Parameters pane.
	3. For the Name, enter extensionLength.
	4. For the Value, enter a number representing the length of the phone number extension.
PL-29815	There was an issue with out of memory exceptions with large data sets when running the phone normalizer upgrade batch process writer. This issue has been fixed. The writer now creates batch process work items in chunks, reducing total heap space required.

Other - Database	
PL-28631	Fixed an issue where SystemToolsAPI.getUpdateStatsState would return the wrong state.
Other - Developer IDE	
PL-27186	Studio now shows warnings for improper usages of internal types when the usage is in an extends or implements clause.
PL-28718	Improvements in IntelliJ with OSGi Editor when you upgrade (or move) a Guidewire product to a new disk location. In IntelliJ with OSGi Editor, open your module settings. Click OSGi Bundle Facet. To the right of Guidewire product directory text field, click the Change button and set the new disk path. The tool updates IDE library dependencies and build.properties.
Other - Integration	
PL-29002	This release adds a new servlet utility class gw.servlet.ServletUtils, which includes methods to get the signed in User object. Use any of three authentication types: 1) the session token from a logged-in user linking from a PCF page; 2) HTTP Basic authentication headers; 3) or name/password pair from custom headers. You can use more than one API in your servlet, for example check the session token, and if it is not available use HTTP Basic authentication. See the Gosu Reference Guide book in the Servlets chapter for details.
Other - Persistence	
PL-28762	Updated the exception for bean overlaps to include more information about the bean including the type of the bean and the fixed id and date range.
Plugins	
PL-28621	Failing to start a startable plugin during server start now prevents the server from advancing to the next run level.
Queries	
PL-27790	Resolved an issue that table aliases in subqueries were not generated with unique values.
PL-28194	With this release, the query builder APIs throw an exception whenever an effdated entity participates in a query through a join statement.
Revisioning	
PL-27210	Fixed an incorrect error that was generated by the Verifies foreign key reference to a subtype is to correct subtype consistency check.
Web - Configuration	
PL-28965	In PCF files, when the LinkCell widget is used under the RowTree widget, the id attribute of the LinkCell widget is now required.
Web - ListViews	
PL-25265	In the PCF widget RowTree, removed support for the pageSize attribute, since the widget is able to resize dynamically. The default value for this attribute is now 0, and any other value produces a validation error.
PL-26720	Added validation to RowTree to prevent using Cell types that cannot be rendered inside the RowTree. For example, CheckBoxCell cannot show check boxes inside the tree. If needed, convert existing usages to the type Cell.
PL-26900	Fixed an issue in ListViews where hidden columns that were redisplayed were not becoming editable when triggered.
PL-28169	Fixed an issue where user-hidden columns caused layout problems by disabling the ability to show/hide columns for columns when a colspan is specified.
PL-28782	Fixed an issue with the RangeValueWidget where RowTree cells were not displaying the proper localized value.
PL-29483	Fixed an issue where grouping a ListView displayed a generic "object" string instead of the correct value.
Web - UI/Runtime	
PL-27761	Fixed an issue where pressing Enter did not properly trigger a search or navigation.
PL-28593	Fixed an issue where the text in the InfoBar was not selectable for copying.
PL-28666	Added validation to the DetailViewPanel configuration to ensure that Input widgets are enclosed by an InputColumn.

PL-28770	Resolved an issue that MonetaryAmountCell did not handle some currency formats correctly.
PL-28854	Fixed an issue with rendering of LinkCells in the RowTree.
PL-28860	You can now localize watermarks in the user interface for date and time fields.
PL-28946	Fixed the DateInput widget for the time to reflect the user's settings for 12-hour versus 24-hour format.
PL-29052	Resolved a critical security vulnerability to non-persistent cross-site scripting attacks.
PL-29475	You can no longer select Group By This Field in a list view when it is in edit mode.
Web Services - WSI (New)	
PL-27467	Fixed an issue where WsiAuthenticationException was being declared in generated WSDLs, but was not actually being returned.
PL-28179	Fixed an issue where WebserviceServlet was waiting for a return to multemode once the runlevel was dropped to maintenance mode before responding to requests.
PL-28267	Web services now throw HTTP 500 for SOAP faults instead of HTTP 200.

Known Issues and Limitations

This topic describes known issues with this release of Guidewire BillingCenter.

Note: For maintenance releases, Guidewire often defers fixing configuration issues that require merging files during the upgrade. Workarounds for many of these issues are listed in the following topics. The goal of this policy is to make upgrades as straightforward as possible.

BillingCenter Known Issues

Rewrite Billing Instruction with the same policy number as the original policy is not incrementing the TermNumber and throws DBDuplicateKeyException (BC-2961)

Issue – If term number "1" already exists for the policy, sending a Rewrite billing instruction without specifying the term number will throw a DBDuplicateKeyException because BillingCenter uses the number "1" for term number.

Workaround – Include a term number that is unique for the given policy number when sending a rewrite billing instruction.

Total Payment Applied on agency bill statement wrong after payment plan edit (BC-2966)

Issue – The Total Payment Applied amount will be incorrect on the next agency bill statement if the following are also true:

- Producer has at least two policies
- Previous agency bill statement is billed and partially paid
- Payment plan used by one of the policies is modified

Workaround – Guidewire is aware of this issue.

Reversing write-off commissions and write-off commission recovery not in the user interface (BC-7184)

Issue – The BillingCenter user interface does not provide a way to reverse write-off commissions or write-off commission recovery.

Workaround – Clone the Reverse button under the Payments tab and reverse CmsnPayableReduction and other transactions using the following action: `CurrentLocation.startEditing()`, `producerTransaction.reverse()`, and `CurrentLocation.commit()`.

Commission: Commission override after a commission write-off causes write-off to be reserved (BC-9536)

Issue – Several methods of the BillingAPI WS-I webservice, such as `issuePolicyPeriod` and `cancelPolicyperiod`, indirectly call `BillingInstructionUtil.executeAndCommit`, which commits the current bundle. These methods also directly call `tryCatch`, which commits a new bundle after executing the entire block of code inside `tryCatch`. As a result, the code tries to commit the same bundle twice.

Following are two ways you can encounter this problem:

Case 1

If you try to create an activity in the block of code passed to `tryCatch` after the call to execute the billing instruction, BillingCenter throws `DBA1readyExecutedException`.

For example:

```
function issuePolicyPeriod(issuePolicyInfo : IssuePolicyInfo, transactionId : String) : String{
    require(issuePolicyInfo, "issuePolicyInfo")
    var publicID = tryCatch( \ bundle -> {
        var pid= issuePolicyInfo.executeIssuanceBI()
        // code *here* to create a new activity
        return pid
    }, transactionId)
    return publicID
}
```

Case 2

If you used `CreateTroubleTicketHelper` to create a trouble ticket in a billing instruction preupdate rule, you would get a `DBA1readyExecutedException`. The following is an example of a billing instruction preupdate rule that would trigger the error:

```
// create trouble ticket with a delq hold until second invoice
var troubleTicketHelper = new CreateTroubleTicketHelper()
var ticket = troubleTicketHelper.createTroubleTicket(
    billingInstruction as PlcyBillingInstruction).PolicyPeriod)
ticket.Title = "Delinquency hold on issuance"
ticket.DetailedDescription = ticket.Title
ticket.Priority = "normal"
ticket.TicketType = "automatic"
ticket.Hold.setAppliedToHoldType(HoldType.TC_DELINQUENCY, true)
ticket.Hold.setReleaseDate(HoldType.TC_DELINQUENCY, gw.util.GWBaseDateEnhancement.CurrentDate)
```

Workarounds

- For Case 1, in the block of code passed to `tryCatch`, do not create or update entities after the call to execute the billing instruction. For example:

```
function issuePolicyPeriod(issuePolicyInfo : IssuePolicyInfo, transactionId : String) : String{
    require(issuePolicyInfo, "issuePolicyInfo")
    var publicID = tryCatch( \ bundle -> {
        // code *here* to create a new activity
        var pid= issuePolicyInfo.executeIssuanceBI()
        return pid
    }, transactionId)
    return publicID
}
```

- For Case 2, instead of using `CreateTroubleTicketHelper` to create the trouble ticket, create related entities directly as in the following example in a PolicyPeriod enhancement:

```
var ticket = new TroubleTicket(this)
var ttJoinEntity = new TroubleTicketJoinEntity(ticket)
ticket.Title = ticketTitle
ticket.DetailedDescription = "Delinquency hold placed on Account with a " + ticket.Title
ticket.Priority = "normal"
ticket.TicketType = ticketType
ticket.Hold.setAppliedToHoldType( HoldType.TC_DELINQUENCY, true )
ticket.Hold.setReleaseDate(HoldType.TC_DELINQUENCY, releaseDate)
ticket.addToTroubleTicketJoinEntities(ttJoinEntity)
this.addToTroubleTicketJoinEntities(ttJoinEntity)
this.Account.addToTroubleTicketJoinEntities(ttJoinEntity)
ticket.Hold.checkForHoldAdditions()
```

PolicyCenter integration on renewal offer: suspense payment is not being applied to policy period (BC-9691)

Issue – When BillingCenter is integrated with PolicyCenter, a suspense payment that is created with an offer number from a PolicyCenter renewal is incorrectly applied to the owner account of the policy period. Correct behavior would be to apply the suspense payment to the renewal policy period instead of the owner account.

Workaround – Guidewire is aware of this issue.

Payment Distribution: BillingCenter incorrectly distributes payment amounts that have been disbursed (BC-9783)

Issue – Reversing a DirectBillPayment is causing T-account unapplied to become negative. This can happen when another direct bill payment on the same account gets modified.

Workaround – Create a recapture charge when unapplied goes below zero.

Net Owed Amount calculation for promises is sometimes incorrect (BC-9920)

Issue – The Net Owed amount is calculated incorrectly for invoice items that have been fully or partially promised.

Workaround – Guidewire is aware of this issue.

Saved promises are listed in two places after upgrade (BC-10019)

Issue – After upgrade, saved promises are also listed under Executed Promises location group.

Workaround – Execute saved promises prior to upgrade.

Inconsistency in behavior when an invoice item (with item level commission rate override) is removed from a charge (BC-10368)

Issue – When an invoice item with item-level commission rate override is modified, BillingCenter recalculates commission inconsistently depending on how the invoice item is changed.

For example, you can modify an existing invoice item on a charge. Alternatively, you can add a new invoice item to the charge and delete the existing invoice item. BillingCenter will calculate the resulting commission differently for these two cases.

Workaround – Guidewire is aware of this issue.

AccountImpl.getNotFullyAllottedFundsTrackers is not bundle safe (BC-11558)

Issue – Manipulating bundles for funds tracking should be avoided. All linked entities are not automatically removed when you remove entities from the bundle.

Workaround – Guidewire is aware of this issue.

Payment redistribution behavior in payment plan refactor changed (BC-12416)

Issue – In BillingCenter 7, payment plan refactor redistributed the entire payment. This behavior has changed as follows:

- In BillingCenter 8, payment plan refactor redistributes only the amount of the payment that had been originally allocated.
- In BillingCenter 7, payment plan refactor respects the account's distribution method. In BillingCenter 8, it does not because an account has no distribution method.

Workaround – If you would like to maintain BillingCenter 7 behavior, contact Guidewire BillingCenter Product Management.

[Upgrading from BillingCenter 3.0.5 or earlier to 8.0 can result in policy commission errors \(BC-13101\)](#)

Issue – In BillingCenter 3.0.5 and earlier, transferring a policy from one producer to a different one and then back to the original corrupted the data by creating duplicate policy commissions. Part of the fix involved an upgrade trigger that consolidated these policy commissions, but this upgrade did not redistribute the commission correctly to the individual invoice items afterward.

Workaround – If you are upgrading from BillingCenter 3.0.5 or earlier, before upgrading to BillingCenter 8.0, first do the minor upgrade to BillingCenter 3.0.6 or a later 3.0 release. After you do the upgrade to BillingCenter 8.0, contact Customer Support for SQL statements and Gosu code that fix any incorrect invoice items resulting from incorrectly redistributed commissions.

[Direct Bill Payment screen does not show the Override Distribution option when columns are hidden \(BC-13455\)](#)

Issue – The Direct Bill Payment screen does not show the Override Distribution option when one or more screen columns are hidden.

Workaround – Show the hidden columns to show the Override Distribution option.

[BillingAPI previewInstallmentsPlanInvoices method shows installments as deposits \(BC-13639\)](#)

Issue – The BillingAPI web service method `previewInstallmentsPlanInvoices` is called by PolicyCenter when issuing a policy. When a Down Payment of 0% is specified for the policy, the installments are incorrectly shown as deposits.

Workaround – Make the following changes.

1. Update the `previewInstallmentsPlanInvoices` method in the `gw.webservice.policycenter.bc801.BillingAPI` web service to remove the `changeEarliestItemPreviewToDownPayment` method call.
2. Replace the `createInvoicesSummary` method in the `gw.webservice.policycenter.bc801.BillingInstructionInfoEnhancement` enhancement with the following:

```
function createInvoicesSummary(bi : PlcyBillingInstruction) : InvoiceItemPreview[] {
    final var itemAmountsByPartitionMap = bi.PolicyPeriod.InvoiceItems
        .partition(\ item ->
            new gw.util.Pair<java.util.Date, InvoiceItemType>(item.InvoiceDueDate,
                item.Type != TC_DEPOSIT and item.Type != TC_INSTALLMENT
                    ? TC_INSTALLMENT : item.Type)
        )
        .mapValues(\ partitionItems -> partitionItems.sum(bi.Currency, \ item -> item.Amount))

    return itemAmountsByPartitionMap.Keys.map(\ partitionKey ->
        new InvoiceItemPreview(null, partitionKey.First, null,
            itemAmountsByPartitionMap[partitionKey], partitionKey.Second)
    ).toTypedArray()
}
```

[Reversing a ProducerPayableTransferTxn may generate inaccurate producer statements \(BC-13821\)](#)

Issue – When using Gosu to reverse a `ProducerPayableTransferTxn` entity, expected behavior is for the transaction to be placed on the next planned statement of each producer. However, depending on the timing of the reversal, the transaction for the receiving producer may be incorrectly placed on a statement that has already been sent. If this occurs, the Previous Balance of subsequent statements will contain inaccurate values.

To determine if your database has been affected by this issue, run the following query:

```
select count(*)  
from bc_producercontext ctx, bc_producerstatement stmt  
where ctx.PayableReceiverStatementID = stmt.ID  
and ctx.CreateTime > stmt.StatementDate
```

Workaround – If this query returns a non-zero count, contact Guidewire Customer Support for assistance.

When rewriting policy in PolicyCenter, selecting a different payment plan is ignored by BillingCenter (BC-13825)

Issue – Integration from PolicyCenter 7 to BillingCenter 8.0.2: When rewriting a policy in PolicyCenter, the selection of a different payment plan is ignored by BillingCenter. BillingCenter retains the original payment plan.

Workaround – In Studio, load the Gosu enhancement RewriteInfoEnhancement contained in the gw.webservice.policycenter.bc700 package. Note: Be sure to load the version in the bc700 package, not the version defined in the bc801 package. In the initPolicyPeriodBIIInternal function, add the following statement at any location:

```
rewriteBI.PolicyPaymentPlan = this.findExistingPaymentPlan()
```

New contact created in BillingCenter for account created in PolicyCenter causes exception (PC-23873)

Issue – When BillingCenter creates a new Contact for an Account that had been created in PolicyCenter, PolicyCenter does not recognize the Contact and throws a BadIdentifierException.

Workaround – Guidewire is aware of this issue.

Studio/Platform Known Issues

Renaming method or property throws ParseResultsException (PL-16633)

Issue – If you rename a property or a method or you change a method signature, and a workflow references that property or method in a Gosu field, BillingCenter throws ParseResultsException. This is the intended behavior.

Workaround – Restart the workflow engine. To do so:

1. Log in to BillingCenter using an administrative account.
2. Access Internal Tools → Reload.
3. Click Reload Workflow Engine.

Javadoc command does not generate index file in expected location (PL-27679)

Issue – When you run gwbc regen-java-api, an index.html file is not created in BillingCenter/java-api/doc.

Workaround – The command now generates Javadoc JAR files in BillingCenter/java-api/doc. To view the Javadoc, add the JAR files to the Studio project.

Long text in table cells can add white space to the right of the page (PL-28288)

Issue – In an editable list view, extremely long text entered in a single cell can cause additional white space on the right side of the page. Long text is text that occupies approximately the entire width of the screen. This issue occurs primarily in Chrome.

Workaround – If you expect users to enter large amounts of text into the cells of a column, configure the column to support text wrapping.

Build tool regen-java-api can fail with -Ddeprecated=true (PL-28992)

Issue – If you run the `gwbc regen-java-api` command with the flag `-Ddeprecated=true`, the command can fail.

Workaround – Specify `dev-deploy` as the value of the `depends` attribute, as follows:

1. Open the `build.xml` file in `BillingCenter/modules/ant` in an editor.

2. Find the entry that starts as follows:

```
<target name="regen-java-api" depends="init"
```

3. Change the value of the `depends` attribute to `dev-deploy`, as follows:

```
<target name="regen-java-api" depends="dev-deploy"
```

4. Save the file and then run the `gwbc regen-java-api` command.

Administrative command-line tools cannot refresh WSDL (PL-29021)

Issue – Some administrative command-line tools rely on web service implementation classes, such as `MaintenanceToolsAPI.gs`. Source files for these classes use the `@Export` annotation, which allows you to edit the file. In this release, the administrative command-line tools cannot refresh the WSDL for these classes. Any change to the web service implementation class that changes the WSDL can prevent the administrative tools from working. Therefore, the only changes you can make to these classes are changes that do not affect the WSDL. For example, you can add `@WsiPermission` annotations to change the permissions without changing the WSDL.

Workaround – Guidewire is aware of this issue.

No entity changed event fired through a one-to-one entity relationship (PL-26224)

Issue – The existence of a one-to-one entity relationship between two entities prevents an entity change event from being fired, even if it is supposed to. For example, in `ContactManager` there is a foreign key from `ABContact` to `Address` for the primary address, and a one-to-one relationship from `Address` to `ABContact`. Because of the one-to-one link, no event changed event is fired for any change in the primary address.

More generally, if entity E1 has a one-to-one relationship to entity E2, no entity changed event is fired for E2 if a property on E1 changes.

Workaround – You might be able to take advantage of a secondary change caused by the initial change, with the secondary change triggering the event to be fired instead. In the previous `ContactManager` example, making sure that changes to all properties in `Address` cause a `History` record to be generated ensures that the entity changed event is fired.

Chrome browser cannot display product documentation in HTML format (DOC-7251)

Issue – If you use the Google Chrome browser, you can view the HTML Guidewire product documentation only if it is served by an HTTP server using the `http://` protocol. The Chrome browser cannot load HTML product documentation from your local disk by using the `file://` protocol.

Workaround – Use a different browser, such as Microsoft Internet Explorer. If you use the HTML documentation only to access the PDF files, you can open them directly in the `pdf` subdirectory of the `doc` directory.

New mechanism for reloading Gosu classes (DOC-8218)

Issue – In past releases, you could modify your PCF files and Gosu classes in Studio, and then reload the changes into the running server by pressing `Alt+Shift+L` in the application user interface. This shortcut no longer loads Gosu classes.

Workaround – Have the server reload your Gosu classes, in Studio, click **Build** → **Make Project**. When Studio is finished compiling your project, the changes will be loaded. You can also restart your server to load the Gosu classes.

[Database Consistency Checks - The date parameter to check against is not correct \(PL-30098\)](#)

Issue – Database Consistency Checks that check date parameters compare against an incorrect date.

Steps to reproduce the problem:

1. Start BillingCenter 8.0.2 on a fresh database.
2. Run database consistency checks. This reports zero failures.
3. Create records for entities that store the datetime when they were created.
4. From server tools, run consistency checks again. Consistency check failures related to the timestamp are reported.

The creation date of an object cannot be after the time. The consistency checks fail because the creation time is compared to the time of the first consistency check run of the application server session instead of to the current database time.

Workaround – Ignore database consistency check errors caused due to timestamp. Or, restart the application server and run consistency checks for the results to be accurate.

[Some PCF widgets in ClaimCenter are miscategorized in Studio \(PL-23454\)](#)

Issue – Some PCF widgets in ClaimCenter are miscategorized in Studio. This means that on certain screens (such as wizard screens), widgets that you may want to use are not available in the toolbox.

Workaround – Find a usage of the missing widget on a different screen, and then copy and paste it to the screen that you are editing.

[Validation for WS-I Web Services not invoked by Studio \(PL-29136\)](#)

Issue – Studio does not validate WS-I web service implementation classes.

Workaround – Before deploying new web service code, run the `gwbc verify-resources` tool from the command line.

[Custom inbound integration implementations require OSGi \(PL-28956\)](#)

Issue – Custom implementations of the `InboundIntegrationStartablePlugin` or `InboundIntegrationMessageReply` plugin interface must use OSGi.

Workaround – Implement the plugin interface as an OSGi plugin.

[JMS inbound integration connectivity problem \(PL-30199\)](#)

Issue – The JMS inbound integration APIs throw exception “JMS Integration Service not supported on server runtime”.

Workaround – Guidewire is aware of this issue.

[Screen elements might disappear when using the AddressAutoFillInput widget \(PL-29653\)](#)

Issue – Quick use of the `AddressAutoFillInput` widget might lead to a JavaScript error, which causes screen elements to disappear.

Workaround – Refresh the page, and the elements will reappear.

Data might disappear while typing while a PostOnChange is firing (PL-30055)

Issue – While a PostOnChange is occurring, the screen may continue to appear responsive, but action or keystrokes captured during that time will be erased when the post returns. This is most noticeable on slow networks or with a large amount of processing occurring during the post.

Workaround – Evaluate your users' network speeds to determine how likely it is that this issue will impact them. This is most likely to affect screens with quick, expert data entry. To improve the speed of the post, set a narrow target for postOnChange, or set it to DATA_ONLY if the user interface does not need to change.

Minute selection in time picker resets to current time (PL-30160)

Issue – When you open a date/time picker, the minutes are reset to the current time's minutes. For example, if the current computer time is 12:37 and the currently set time in the picker is 5:50, then opening and closing the picker changes the time to 5:37 and marks it as an edit.

Workaround – Guidewire is aware of this issue.



chapter 19

Guidewire BillingCenter 8.0.1 Release Notes

Release 8.0.1

Document Revision: 02-July-2015

Copyright © 2001-2015 Guidewire Software, Inc. All rights reserved.

Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire Reinsurance Management, Guidewire ContactManager, Guidewire Vendor Data Management, Guidewire Client Data Management, Guidewire Rating Management, Guidewire InsuranceSuite, Guidewire ContactCenter, Guidewire Studio, Guidewire Product Designer, Guidewire Live, Guidewire DataHub, Guidewire InfoCenter, Guidewire Standard Reporting, Guidewire ExampleCenter, Guidewire Account Manager Portal, Guidewire Claim Portal, Guidewire Policyholder Portal, Gosu, Deliver Insurance Your Way, and the Guidewire logo are trademarks, service marks, or registered trademarks of Guidewire Software, Inc. in the United States and/or other countries.

All other trademarks are the property of their respective owners.

This material is confidential and proprietary to Guidewire and subject to the confidentiality terms in the applicable license agreement and/or separate nondisclosure agreement.

Guidewire products are protected by one or more United States patents.

Overview

These release notes contain the following topics:

- Release Information
- Installing This Release
- Support
- Issues and Major Changes
- Improvements and General Issues
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire BillingCenter.

IMPORTANT If you skipped one or more upgrade releases to BillingCenter, be sure to read the Release Notes for those releases to learn about changes and fixed issues.

Version Number

This release of Guidewire BillingCenter is 8.0.1.

- If you are integrating with Guidewire PolicyCenter, this version of BillingCenter requires PolicyCenter 7.0.6 or later. The latest maintenance release is preferred.
- If you are integrating with Guidewire ContactManager, this version of BillingCenter requires ContactManager 7.0.0 or later. The latest maintenance release is preferred.

Installing This Release

Refer to the following:

- The *BillingCenter Installation Guide* for general installation information
- Prior BillingCenter release notes for any versions that you have skipped

Support

For assistance, visit the Guidewire Resource Portal – <http://guidewire.custhelp.com>

Issues and Major Changes

This topic contains issues and major changes that can affect your installation. For information on new features and major changes, see the topic “New and Changed in BillingCenter 8.0.1” in the BillingCenter New and Changed Guide.

This topic includes:

- Upgrade Support
- Base PCF File Changes
- Changes in This Release Provided in Upgrade Diff Report

Upgrade Support

Upgrading from BillingCenter 3.0.4 or newer versions is supported in this release.

The following information will be added to the *Upgrade Guide* in future releases:

Upgrading from BillingCenter 2.1 to BillingCenter 8 requires upgrading to BillingCenter 7 first

BillingCenter 8 supports direct upgrade only from BillingCenter 3 and BillingCenter 7.

To upgrade to BillingCenter 8 from BillingCenter 2.1, you must:

1. Upgrade to BillingCenter 7.

2. Run the following work queues:

```
com.guidewire.bc.domain.commission.impl.UpgradeCommissionWorkQueue  
com.guidewire.bc.domain.accounting.billinginstruction.impl.UpgradeBillingInstructionPayme  
ntPlanWorkQueue
```

3. Upgrade to BillingCenter 8.

Upgrade fails if BCContact and PaymentDetails have customer extensions

If you are upgrading from BillingCenter 3 or earlier, and you have added extension fields added to the BCContact or PaymentDetails entities, upgrade does not migrate data in those fields to BillingCenter 8.

Before you upgrade, remove any extension fields from BCContact.etx and PaymentDetails.etx and add those extension fields to the corresponding supported entity. The extension field names and definitions must not change.

- Move any extension fields from BCContact.etx to Contact.etx.
- Move any extension fields from PaymentDetails.etx to PaymentReceipt.etx.

Note: You must not change the name or definition for each field you migrate from the old entity extension to the new entity extension.

Upgrade does not remove data rows for entities that have changed from retireable to non-retireable

In cases in which the definition of an entity has been changed from being retireable to being non-retireable, BillingCenter 7.0.3 and later versions does not automatically remove data rows that have been retired. Instead, before upgrading to BillingCenter 8.0.1, a new version check,

MigrateTableFromRetireableToEditableVersionCheck, runs. This version check logs an error if any retired rows are found. If any such errors are logged, you must examine and remove the retired rows or contact Guidewire Customer Support for assistance with restoring the retired rows.

Base PCF File Changes

The link that follows requires that the ReleaseNotes_files directory be on your local disk in the same directory as this release notes file.

BillingCenter release 8.0.0 to 8.0.1

To view a report of the changes to the base PCF files, [click here](#).

Changes in This Release Provided in Upgrade Diff Report

Guidewire provides a report detailing certain differences between the current release and your prior release. This report describes changes in display keys, entities, typelists, and the Gosu API. To obtain your custom Upgrade Diff Report, visit the Guidewire Resource Portal.

Improvements and General Issues

This topic describes improvements and issues corrected in this release. Guidewire attempts to provide information for issues of primary importance to our customers. This list is not intended to be comprehensive.

ID	Description
ABIN – ContactCenter	
BC-12873	Updated the ABContactSystemPlugin class to handle error messages that ContactManager can return that are based on the locale of the user in BillingCenter.

ACCT – Account

BC-12990	A call to <code>BillingAPI.transferPolicyPeriods</code> , which transfers policy periods to a different account, now ensures that policy periods are transferred to the sibling account with the correct currency. If such an account doesn't already exist, the method creates one.
BC-13033	Funds in any unapplied fund can now be adjusted. The existing API <code>Account.makeAdjustment</code> adjusts the default unapplied fund. To adjust a different unapplied fund, use <code>UnappliedFund.makeAdjustment</code> . Additionally, you can now use <code>PaymentAPI.makeAccountAdjustment</code> to adjust a default or a designated unapplied fund.
BC-13170	The Account creation API can now specify a payment allocation plan on account creation. If not specified, the account is linked to the payment allocation plan that has the highest plan order. To support this feature, <code>PCAccountInfo</code> now has the field <code>PaymentAllocationPlanPublicID</code> . Additionally, <code>BillingAPI</code> now has the method <code>findAllPaymentAllocationPlans</code> , which returns a selection of payment allocation plans.

ACCT – Account, PMPN – Payment Plan and Invoicing

BC-9073	Corrected the error message shown in BillingCenter when the account type of an existing account is changed.
BC-13001	<p>The <code>AccountDistributionLimit</code> no longer exists on <code>Account</code>. This functionality is now part of the account's associated payment allocation plan. Additionally, the <code>DistributionLimitType</code> field on <code>Account</code> is now deprecated and is not used in the base configuration of BillingCenter. You also no longer see the account distribution limit setting on the <code>Account → Summary</code> screen, <code>AccountDetailSummary.pcf</code>.</p> <p>The Distribution Criterion filters on payment allocation plans are restrictions that determine which invoice items can be targeted for payment. Three of these filters reflect the functionality previously provided by the account distribution limit. They are <code>BilledOrDueDistributionFilterCriterion</code>, <code>NextPlannedInvoiceDistributionFilterCriterion</code>, and <code>PastDueDistributionFilterCriterion</code>. If you do not configure any of these filters, the default is equivalent to <code>Amount Up to Under Contract</code>.</p> <p>You can set these filters in the New Payment Allocation Plan screen by navigating to Administration tab -> Actions -> New Payment Allocation Plan. Alternatively, you can edit an existing payment allocation plan by navigating to Administration tab -> Business Settings -> Payment Allocation Plans. The default filter for a new payment allocation plan is <code>Billed or Due</code>. Additional filters available are <code>Next Planned Invoice</code> and <code>Past Due</code>.</p>

ADMN – Administration

BC-12955	When you create and edit custom authority limit profiles, they now fetch the correct authority limits.
----------	--

AGBL – Agency Bill

BC-12511	<p>There was a performance problem when adding statements to an <code>Agency Bill Payments</code> screen or an <code>Agency Bill Promises</code> screen by using search. This problem has been fixed by adding the following denormalized columns on <code>InvoiceItem</code>: <code>CanBePaidMoreByAgencyBill</code> and <code>CanBePromisedMoreByAgencyBill</code>. These fields are true only if the invoice item is under-settled. Specifically:</p> <ul style="list-style-type: none"> The <code>CanBePaidMoreByAgencyBill</code> field is true for positive items for which either of the following is true: $((\text{paid gross} + \text{written off gross}) < \text{gross amount})$ or $((\text{primary active paid commission} + \text{earned commission} + \text{written off commission}) < \text{commission amount})$. If the invoice item is exactly settled or over-settled in both its gross and primary active commission, then this field is false. The <code>CanBePromisedMoreByAgencyBill</code> field simply evaluates the following promised amounts: $((\text{paid gross} + \text{written off gross} + \text{promised gross}) < \text{gross amount})$ or $((\text{primary active paid commission} + \text{earned commission} + \text{written off commission} + \text{promised commission}) < \text{commission amount})$.
----------	---

AGBL – Agency Bill, PERF – Performance (INTERNAL)

BC-12823	<p>There was a performance issue that occurred with adding invoice items to an agency bill payment with very large numbers of invoice items. This issue was a result of the way the search criteria were cached. This issue has been fixed by adding a Gosu object, <code>gw.search.InvoiceItemSearchCriteria</code>, which replaces the non-persistent entity <code>InvoiceItemSearchCriteria</code>. The entity <code>InvoiceItemSearchCriteria</code> has been deprecated but not removed.</p> <p>If you are upgrading to BillingCenter 8.0.1 and this performance issue has not previously affected you, you can continue to use the entity version of the search criteria. In this case, do not merge the changes in <code>InvoiceItemSearchPanelSet.pcf</code>, and replace references to <code>gw.search.InvoiceItemSearchCriteria</code> with <code>entity.InvoiceItemSearchCriteria</code>.</p>
----------	--

Application Server	
PL-27203	JBoss 6 deployment now works properly.
Archiving	
PL-26930	Modified the <code>restoreSeveredTransactionOffsetOnsetOnsetLinks</code> method so it can be called from Gosu.
AUDT – Audit Process	
BC-13160	BillingCenter can now issue final audits in a non-default currency.
Batch Processes	
PL-18406	<p>In releases prior to 8.0.0, you created custom batch processes by developing Gosu classes that implement the <code>BatchProcess</code> interface and subclass from class <code>BatchProcessBase</code>. You configured custom batch processes in the <code>scheduler-config.xml</code> file to run on a schedule or on demand from the Administrative user interface. Custom batch processes ran only on the batch processing host in the cluster.</p> <p>Beginning with release 8.0.0, you also can create custom batch processes by developing Gosu classes that subclass from the <code>WorkQueueBase</code> class. You configure this newer type of custom batch process in the <code>work-queue.xml</code> file, with definitions for its worker processes that run on other servers.</p> <p>Note: The methods on <code>WorkQueueBase</code> are called from different threads at different times on different servers. So, do not attempt to use class properties to configure or maintain state.</p>
BLPN – Billing / Billing Plan	
BC-4285	The <code>DunningInterval</code> property has been removed from the <code>Plan</code> entity because it is not used in the base configuration. This change affects all subtypes of <code>Plan</code> , such as <code>BillingPlan</code> , <code>AgencyBillingPlan</code> , <code>DelinquencyPlan</code> , <code>PaymentPlan</code> , and so on. The <code>Dunning Interval</code> field has also been removed from the Billing Plan detail and edit screens, and all supporting consistency checks, display keys, and so on have also been removed. If you are upgrading from an earlier version of BillingCenter and your configuration uses this property, consider implementing <code>DunningInterval</code> as a script parameter.
BC-13111	Changing plan order across an order gap no longer throws a <code>NullPointerException</code> .
Bundles and Transactions	
PL-22240	The <code>touch</code> method has been re-implemented. Please refer to the Gosudoc for detailed usage.
CHRG – Charge/Charge Group	
BC-1225	<p>There was a problem with the order of payment of recapture charges that has been fixed. Recapture charges now have a higher priority for payment distribution in the base configuration.</p> <p>Prior to this fix, recapture charges behaved as follows:</p> <ul style="list-style-type: none"> • When a recapture charge was created, there was no impact to unapplied. • When a recapture charge was paid, it credited back the owner's unapplied. <p>Recapture charges now behave as follows:</p> <ul style="list-style-type: none"> • When a recapture charge is created, the unapplied is credited right away. • When the charge is paid, unapplied is not changed.
CLAG – Collection Agency, SECU – Security (Login, Permissions/Roles)	
BC-11175	Added the role Direct Collector to the sample data. In the base configuration, this role has no permissions associated with it.
CMD – Command Line Tools	
BC-10592	The RPCE implementations <code>IMaintenanceToolsAPI</code> and <code>IMessagingToolsAPI</code> have been deprecated. Use the new WSI implementations <code>MaintenanceToolsAPI</code> and <code>MessagingToolsAPI</code> instead.
CMSN – Commission / Commission Plan	
BC-10816	Changed the Commission Subplan worksheet card Commissionable Items to show only charge patterns with T-account owner Policy Period and charge category Premium. By default, the value of the Item field is Premium. This card no longer shows any type of fee.

BC-13180	<p>Fixed a problem that left invoice items in an inconsistent state if they were not planned (for example, they were billed) and:</p> <ul style="list-style-type: none"> They had associated commission but no associated payment, such as an unpaid item that has commission because of pay on bind. They were individually assigned to another account or producer <p>In this case, BillingCenter creates an onset invoice item. The producer transactions associated with the original item are now getting transferred to the onset.</p>
----------	--

CMSN – Commission / Commission Plan, PMPN – Payment Plan and Invoicing

BC-218	Fixed a problem that caused commission not to be earned when a previously inactive producer code was made active.
BC-11976	Fixed a problem with payment redistribution after a payment plan refactor. Payments now distribute across commissions even when the gross is zero.

Cognos Integration

PL-21153	Thread usage on the LDAP server is now managed more efficiently.
----------	--

Command Line Tools

PL-27378	Removed debug-start from command line option.
PL-28544	The obsolete command line command copy-theme has been removed.

Configuration Upgrade

PL-27207	Log file of upgrade tool now reflects the difference between platform upgrade triggers and application-specific upgrade triggers.
PL-27208	You can now expand all and collapse all in the upgrade tool.
PL-27209	You can now right-click to export the directory tree within the upgrade tool.
PL-27338	Fixed an issue in which running upgrade multiple times created nested rules folders.
PL-27744	Fixed an issue that occurred if you added any ClaimContactInput widgets that use postOnChange, or if you modified a base ClaimContactInput widget by adding postOnChange. In these cases, the postOnChange would not get upgraded to the new syntax.

Consistency Check

BC-12864	<p>If your installation has very large databases, it can be expensive to run consistency checks. Guidewire performance testing found that some of the queries used in the consistency checks used a large amount of temporary space on Oracle databases.</p> <p>This fix provides configuration parameters that you can set in config.xml. These new configuration parameters enable you to split some of these queries into multiple smaller queries, each of which checks only a portion of the database. The higher the value you set, the larger the number of queries used, and the smaller each one is. You are not likely to need to set these configuration parameters unless you have very large databases.</p> <p>The new configuration parameters are:</p> <ul style="list-style-type: none"> MaxTransactionsInConsistencyCheck - Default value is 0, meaning do not split. MaxTAccountsInConsistencyCheck - Default value is 0, meaning do not split.
----------	---

CONT – Contacts

BC-12685	There was a problem with the Account Summary screen showing the incorrect primary contact when more than one contact had the role Insured. This problem has been fixed, and now the Account Summary screen shows the primary payer as the account's contact.
----------	--

CRED – Ad Hoc Credit

BC-11025	<p>A new typekey, ListBillAccountExcessTreatment, has been added to the ReturnPremiumPlan entity. This property targets excess returned premium to the owning account's Unapplied when the payer account is ListBill. The typekey gets its value from the ListBillAccountExcess typelist, which in the base configuration has the following typecodes:</p> <ul style="list-style-type: none"> PolicyPayerUnapplied – Specifies the policy payer unapplied as the target for any returned premium. PolicyOwnerUnapplied – Specifies the policy owner default or policy designated unapplied as the target for any returned premium.
----------	--

Data Distribution	
PL-25702	The Data Distribution page enables Download comparison Zip file and Download Combined Zip file options only if two executions are selected. In prior versions, the options were erroneously enabled even if there were not two executions selected.
Database Configuration	
PL-28656	Fixed a problem that prevented the DBAuthenticationPlugin from working.
Database Support	
PL-24417	The performance requirement to set the <code>action="delete"</code> attribute for <code>bc_message</code> in <code>database-config.xml</code> has been removed.
PL-27438	Added support for Oracle Date interval partitioning.
PL-28323	Fixed a bug which prevented scheduling of the Database Statistics process.
Database Support – Oracle	
PL-26203	Nullable <code>monetaryamount</code> columns will use the Oracle default value feature during upgrade. This is a performance optimization.
PL-28085	The Oracle fast add column feature is used when adding <code>monetaryamount</code> columns to an entity or a subtype.
PL-28327	Guidewire added an option to switch off the Oracle adaptive optimization feature for Guidewire applications. See “Configuring Oracle Adaptive Optimization for BillingCenter” in the <i>BillingCenter Installation Guide</i> .
Database Upgrade	
PL-27016	Guidewire has added support for using Oracle's parallel DDL execution feature during upgrade. The <code>createIndexInParallel</code> attribute of the <code><upgrade></code> element has been replaced with the new <code>degree-parallel-ddl</code> attribute. See “Configuring Parallel DML and DDL Statement Execution” in the <i>Upgrade Guide</i> .
PL-27865	Guidewire resolved a rare issue in which the server would be able to start after an incomplete upgrade but subsequent upgrade attempts would fail.
PL-27918	The Database Upgrader now honors the database statistics configuration.
PL-28041	The <code>updatestatistics</code> attribute in <code>database-config.xml</code> now controls both deletion and collection of database statistics during upgrade.
PL-28122	The <code>DeferredUpgradeTasks</code> process is now profiled and appears under Guidewire Profiler. This process is used when deferring creation of archive indexes until after the upgrade. See “Deferring Creation of Archive Indexes” in the <i>Upgrade Guide</i> .
PL-28449	Fixed a bug that prevented handling of statistics on the ID column properly.
DBUP – Database Upgrader	
BC-12774	The database upgrade now correctly specifies the localized versions of the name and description fields for the default Return Premium plan and Payment Allocation plan in the base configuration. The names and descriptions created by the database upgrade for these plans are temporary names indicating that the plans were upgraded. You must update these fields with your localized names and descriptions.
BC-12835	There was an issue regarding upgrading an account with the default <code>PaymentMethod</code> set to <code>Wire</code> when upgrading from BillingCenter 3.0. to 8.0. This issue has been fixed as follows: The upgrade for Account and Producer entities has been changed to convert a <code>PaymentMethod</code> to a <code>PaymentInstrument</code> by using the <code>PaymentMethod</code> type. The default <code>PaymentInstrument</code> is set according to the pre-upgrade <code>PaymentMethod</code> . The exception is that disbursements having outgoing payments convert to the <code>PaymentInstrument</code> of that outgoing payment.
BC-12846	The upgrade for Account and Producer entities has been changed to convert a <code>PaymentMethod</code> to a <code>PaymentInstrument</code> by using the <code>PaymentMethod</code> type. The default <code>PaymentInstrument</code> is set according to the pre-upgrade <code>PaymentMethod</code> . The exception is that disbursements having outgoing payments convert to the <code>PaymentInstrument</code> of that outgoing payment.
BC-12904	Fixed an issue in which upgrading from BillingCenter 3.0.x or 7.0.x threw an exception if a <code>BaseMoneyReceived</code> object was being pointed to from multiple <code>BaseDist</code> objects. For example, this exception was happening during upgrade from BillingCenter 3.0 when there were retired <code>BaseDist</code> objects.

BC-13029	<p>AccountDistributionLimitType has been removed in favor of using payment allocation plans. During upgrade, the upgrader looks at the AccountDistributionLimitType objects currently in use by accounts in the system and creates a corresponding payment allocation plan for each one it finds. These plans are then connected to existing accounts based on what limit type they were using.</p> <p>The order for the plans created is based on frequency. The payment allocation plan that reflects the most commonly used distribution limit type has the highest plan order, and thus becomes the default when creating new accounts.</p> <p>For distribution limit types that were not in use, no corresponding payment allocation plan is created. If multiple plans are created, they are given a name that reflects the distribution setting they correspond to. If only one plan is created (meaning that only one distribution type was in use), it will be named Default Payment Allocation Plan.</p>
BC-13185	<p>Remaining issues with PaymentInstrument upgrade have been fixed. Now the following upgrades work properly:</p> <ul style="list-style-type: none"> • Disbursements with outgoing payments • Collateral disbursements with and without outgoing payments for which the payment method is ACH/EFT without the bank name • Account disbursements with and without outgoing payments for which the payment method is Credit Card without the credit card issuer • Suspense disbursements with and without outgoing payments for which the payment method is Wire without the bank name • Agency disbursements with payment made with payment method ACH/EFT with no bank name. Including outgoing payments <p>After the upgrade runs:</p> <ul style="list-style-type: none"> • The payment method upgrades to the correct payment instruments in BillingCenter 8.0.1, even without credit card issuer or bank name. • All payment instruments on disbursements are correct after upgrade. • For accounts, the upgrade converts a PaymentMethod to a PaymentInstrument by using the PaymentMethod type.

DLNQ – Delinquency / Delinquency Plan

BC-12749	There was a problem with the delinquency process consistency check incorrectly failing for policies that had closed delinquencies that were transferred to another account. This consistency check has been changed to look at only active delinquencies, and it no longer fails for this legal situation.
----------	--

DLNQ – Delinquency / Delinquency Plan, PMPN – Payment Plan and Invoicing

BC-13187	The New Policy wizard now limits the currency of the displayed delinquency plan and payment plan drop-down lists to the currency of the policy period.
----------	--

DLNQ – Delinquency / Delinquency Plan, STAG – Staging Table

BC-13168	You can now load account-level delinquencies into BillingCenter by using staging tables. Previously, only DelinquencyProcess rows of subtype PolicyD1nqProcess were allowed. Now the account-level subtype DelinquencyProcess is allowed. In addition, the LegacyDelinquency work queue now traverses both these subtypes after the integrity-check-and-load phase.
----------	---

Document Management

PL-27480	Document templates no longer have size constraints.
PL-27577	You can now call the method DocumentsUtil.createNewDocument without needing the current user to have the doccreate permission. Use this method if you implement the IDocumentMetadataSource plugin.

Email

PL-27921	Due to potential cross-site scripting vulnerabilities, HTML is disabled as the content of an email. If you would like to continue to have HTML email and accept the risk of this vulnerability, you may remove the escaping of the subject and body in the document template.
----------	---

Entities/Metadata

PL-19023	The <i>Data Dictionary</i> has been modified to show references for subtypes
PL-24743, PL-27611	<i>Data Dictionary</i> descriptions for core locale fields have been updated.

PL-25622	Fixed an issue in which MonetaryAmount appeared as two separate fields under Actual Amount in the <i>Data Dictionary Data Entity View</i> .
PL-25809	Fixed an issue in which typelist codes did not include documentation for the typecodes (name and description).
PL-27465	Added upgrade trigger to insert xmlns if missing on .eti files.
PL-27501	Added the ability to override and add keyfilters and typefilters by using extensions.xml.
PL-27819	Added a new overlapTable attribute to the edgeForeignKey and localization entities, which specifies that the entity implements the OverlapTable delegate.

FUND – Funds Tracking

BC-5857	A screen in which you can enable and disable funds tracking has been added to the Server Tools tab. To enable or disable funds tracking, press Alt+Shift+T, and then on the Server Tools tab, click Funds Tracking in the sidebar. Before enabling or disabling funds tracking, be sure to read about this feature in the Application Guide so you understand the consequences.
BC-12869	Fixed a bug in the mapping set up between a FundsSourceType and the corresponding IEntitytype. This code bug resulted in an exception being thrown intermittently when the Allot Funds for Funds Tracking batch process was run.

Globalization

PL-26606	Updated GroupUserSearchDV.pcf to support Japanese kanji fields in the standard way represented elsewhere in the application.
PL-27394	Values substituted in display keys that are of type BigDecimal, Date, and IMoney and its implementing classes, such as MonetaryAmount, are now properly formatted according to the regional formats in effect for users. For example, the substitution value "123456" is formatted as "123,456" for U.S. (English) or "123 456" for France (French).
PL-27609	Fixed an issue in which the display value for locale shifts when an admin user is changing his/her own language/locale when viewing the profile of another user.
PL-28068	The configuration parameter for overriding the default maximum width for labels moved from display key ExtJS.Form.LabelWidth to XML element LabelWidth, with attribute width specified in pixels. Use the LabelWidth element in the language.xml file for the language that you want to configure. For example,
	<pre><GWLanguage code="de_DE" name="German (Germany)" typecode="de_DE"> <ExtJsSettings> <LabelWidth size="220" /> </ExtJsSettings> </GWLanguage></pre>

Gosu

PL-18217	A Gosu class now preserves annotations inherited from Java classes in the class hierarchy. Note this change effectively restricts access by a subclass to features tagged with the @InternalAPI annotation.
PL-25700	Fixed an issue that caused errors on startup under very specific conditions related to the compilation of particular Gosu classes.
PL-27099	Gosu supports annotations on parameters in methods, properties, and constructors.
PL-27428	The Gosu language now provides limited support for the Java annotation @SuppressWarnings, which tells the compiler to suppress warnings. Use this annotation on declarations of a type, function, property, constructor, field, or parameter. Note that local variables do not support this annotation. You must pass a String value as an argument to indicate what warnings to suppress. Pass the argument "all" to suppress all warnings. Pass the argument "deprecation" to suppress deprecation warnings. For example, to suppress deprecation warnings in a Gosu class, add the annotation @SuppressWarnings("deprecation") on the line before the class declaration.
PL-27651	Gosu now recognizes the annotation @java.lang.Deprecated as a form of deprecation, in addition to @gw.lang.Deprecated and the @deprecated Javadoc tag.

I18N - Internationalization

BC-13069	The following field labels on a policy's Change Payment Plan screen are now properly localized: <ul style="list-style-type: none">• Items to Process• Edit Type
To open this screen in BillingCenter, with a policy open, navigate to Payment Schedule → Edit Scheduled .	

Integration

PL-28196	This release changed how to configure inbound multi-threaded integrations such as the built-in file and JMS integrations. In previous releases, you added configuration parameters in the Plugins registry in Studio. In this release, you set a single parameter <code>integrationservice</code> and then do the rest of the configuration in the new file <code>inbound-integration-config.xml</code> . Also, the API details for file and JMS integrations changed. There is a new plugin interface called <code>InboundIntegrationHandlerPlugin</code> . Also, the file integration now supports processing one file at a time, rather than one line at a time. For details, refer to the <i>Integration Guide</i> .
----------	--

IntelliJ IDE – Compiler

PL-28346	Fixed a compilation error when compiling an entity or its extension if the entity had a subtype.
----------	--

IntelliJ IDE – Debugger

PL-27875	When debugging Gosu code, you can now browse the structure of an Entity object and inspect the property values stored within it. To enable this, in Guidewire Studio, click File → Settings , and then in the Guidewire Studio panel set Enhance Entities Visualization .
----------	--

IntelliJ IDE – Display Key Editor

PL-27971	Improved typing performance in the Display Key editor.
PL-28677	Fixed a bug in Studio in which converting a string into a display key caused exceptions. The behavior has changed slightly. Now the locale folder is required to have a <code>display.properties</code> file already in it before it appears in the Create Display Key dialog or in the Step Name Localizations tab in the Workflow editor.

IntelliJ IDE – Entity Editor

PL-26336	Column validation has been re-enabled in Studio.
PL-26364	Fixed an exception in the Entity editor when attempting to override a read-only attribute.
PL-26375	Fixed an issue in the New Entity dialog in which the <code>viewEntity</code> was listing suggestions that were not applicable.
PL-26540	Added additional error notes to the Entity editor to highlight the parent elements if a child is invalid.
PL-27220	In the Entity editor in Studio, when editing the <code><tag></code> subelement of the <code><column></code> element, there is now a drop-down list showing available values.
PL-27454	Fixed an issue in which creating multiple entity extensions with suffixes would produce an exception.
PL-27715	Fixed an error that would occur when the <code>effDatedBranchType</code> attribute was not correct in an entity of type <code>effdated</code> .
PL-28089	In the Entity editor in Studio, you are now required to specify a value for the <code>nullok</code> attribute.
PL-28529	Fixed an error in Studio when creating an entity extension if there is a Java class under <code>src</code> .

IntelliJ IDE – Gosu Editor

PL-19418	Fixed a compilation error during bytecode generation on certain annotations from Java source types.
PL-26640	Fixed an exception that would occur when typing "display" into a Gosu class.
PL-26866	The Gosu <code>using</code> clause syntax now has an additional feature for adding additional cleanup code. You can optionally add a <code>finally</code> clause that runs after the statement body, even if exceptions occur in the body of the <code>using</code> clause. See the <i>Gosu Reference Guide</i> for details.
PL-27135	Studio now shows additional warnings for improper usages of internal gw classes.
PL-27320	Fixed an issue in which pressing <code>Ctrl+0</code> threw an exception in Gosu.
PL-27724	Fixed an issue with some deprecated methods not being shown in strikethrough text in Studio.
PL-27873	Fixed an issue with pasting code into Studio in which multiline statements were concatenated and merged with comments.
PL-27893	Fixed an exception that was thrown when creating a new Gosu template.
PL-27943	Fixed an issue with some deprecated methods not being shown in strikethrough text in Studio.

PL-27944 The Gosu language has two new compound assignment operators, which are operators that apply an operation to a variable and then re-assign the variable to the result. The new operator `&&=` performs the logical AND operation to the previous value. The new operator `||=` performs the logical OR operation to the previous value. Both operators work with the primitive type `boolean` or the object type `Boolean` on either side of the operator. For example, suppose you have two `boolean` variables called `needsUpdate` and `flagTest`. The statement `needsUpdate ||= flagTest` has the meaning `needsUpdate = (needsUpdate OR flagTest)`. Do not confuse these new operators with the other operators `&=` and `|=`, which apply bitwise AND and bitwise OR operations.

PL-28019 Fixed a false compile error in the Gosu editor that manifested when a property getter or setter overrode a getter or setter in a superclass implemented in Java.

PL-28027 Gosu does not support numeric expressions in the for statement after the `in` keyword. The following code is illegal:

```
for (x in 10) {...}
```

This illegal code must be upgraded by using the provided inspection in Studio with an interval such as:

```
for (x in 0..|10) {...}
```

IntelliJ IDE – Line of Business Editor

PL-26590 The LOB tab has been removed from .tti files in Studio to prevent missing loss types.

PL-26595 Added the ability to more easily select multiple typekeys in Studio using the Ctrl or Shift keys.

PL-26620 Retired typecodes are now shown in strikethrough text.

PL-26826 Fixed an issue in which Studio would throw an exception when a categorylist was added to a LossType typecode.

PL-26971 Fixed an issue in Studio in which removing a typecode from its parent also incorrectly removed it from all of its other parents.

IntelliJ IDE – Other

PL-27198 Fixed an issue in which Run commands in the QuickJump box did not work when the server was started from Studio.

PL-27862 Fixed an issue in which Studio would not suggest types defined on XSD files when trying to create an enhancement.

IntelliJ IDE – PCF Editor

PL-26516 Improved the PCF editor to highlight the correct panel when selecting widgets in nested files.

PL-27147 The PCFMapping tool has been updated to include fields such as `PanelIterator`.

IntelliJ IDE – Plugins, OSGi, Plugins

PL-27497 You can now implement plugin interfaces in Java using the OSGi standard. OSGi is a Java module system and service platform that helps isolate code modules and any necessary Java libraries. Guidewire recommends OSGi for all new Java plugin development. To simplify OSGi configuration, BillingCenter includes IntelliJ IDEA with OSGi Editor, an application separate from Guidewire Studio. For more information, refer to the *Integration Guide*.

IntelliJ IDE – Refresh

PL-24108 Fixed an issue in which the server threw an exception and did not handle newly created enhancements.

PL-28174 Fixed an issue in which methods added in entity classes were invalid until restarting Studio.

PL-28187 Fixed an error that would occur after renaming an element in an XSD and then navigating to a Gosu type that contained a usage of that element.

IntelliJ IDE – Typelist Editor

PL-24391 Fixed the typelist editor in Studio to filter out options under the drop-down as you enter text.

PL-26445 Fixed an issue in which creating the first extension of a typelist caused an exception.

PL-26613 Fixed an issue in Studio in which disabled typecodes looked enabled when selected.

PL-26535 Fixed an exception in the text editor of the typelist view.

PL-27174 Fixed an issue in which the default setting for filtering metadata did not apply to the Typelist editor.

PL-27570 Fixed an issue in which clicking on the name attribute in a typelist extension would not allow you to override it.

PL-27584 Fixed the Add To Category dialog so retired typecodes appear in strikethrough text as options for filtering.

PL-28177 In Studio, the Entity editor and Typelist editor are now case-insensitive when resolving references to other metadata.

PL-28515 Fixed an issue that generated multiple errors in the Studio Typelist editor in the localization panel.

IntelliJ IDE – Web Services Editor

PL-26576 Added a check to the timeout value of web services to insure that it is lower than Studio's maximum integer.

INTG – General Integration bug, PCIN – PolicyCenter Integration

BC-12814 Fixed a line of business typecode mismatch between PolicyCenter and BillingCenter. This mismatch caused an exception to be thrown when a commercial package policy was created in PolicyCenter.

BC-12946 BillingAPI now supports changing, canceling, or reinstating a policy period that uses a non-default currency. The affected BillingAPI methods are:

- changePolicyPeriod
- cancelPolicyPeriod
- reinstatePolicyPeriod

BC-13067 BillingAPI.updateAccount now throws a BadIdentifierException if the specified account does not exist.

BC-13071 The Javadoc for BillingAPI.createAccount previously stated incorrectly that this method returned a public ID. This method returns an account number.
The Javadoc has been corrected in this release.

BC-13199 A separate fix added a PCPublicID field to the BillingCenter Policy entity. This field was added so that BillingCenter could find a PolicyPeriod by using a unique ID, rather than by using the PolicyNumber field, which might change. For BillingCenter installations that upgrade from version 7.0.0 or earlier, this new field is null.
This fix has to do with the circumstances under which an API call that contains the content for the PCPublicID field might populate the field if it is null. Previously, a call to findPolicyPeriod from the API did not update the new field because findPolicyPeriod was normally, but not always, called from a read-only context. Now, any API call that executes a BillingInstruction, including findPolicyPeriod, updates the new field. API calls associated with a BillingInstruction normally involve making and committing changes.

INV – Charge / Invoice, PADM – Policy Administration / Integration

BC-8637 The premium report billing instruction now respects the payment plan settings and does not automatically issue the entire amount as a down payment. If you are upgrading and want to maintain existing behavior, you might need to change existing integrations by customizing their payment plans.

Miscellaneous / None

BC-5191 The signature for the onChargesPaid method of the IDelinquencyProcessExtensions plugin interface has changed. This method's charges parameter is now a list of type Charge instead of a list of type ChargePaidFromUnapplied.

The signature is now:

```
void onChargesPaid(
    gw.bc.delinquency.entity.DelinquencyProcess delinquencyProcess,
    java.util.List<gw.bc.accounting.entity.Charge> charges);
```

BC-12834 Fixed a problem with BillingCenter throwing an exception if the priority of a cloned plan was moved up. This exception, which no longer occurs, previously occurred if the cloned plan was initially given the name of an existing plan and then renamed.

BC-12858 Several work queues are marked as disabled (`instances="0"`) in the `work-queue.xml` file in the base configuration. These work queues are needed only under special circumstances, such as after an upgrade or after loading a staging table. To improve performance, leave these work queues disabled during normal server use.

Also removed a comment that indicated incorrectly that only one worker instance must be created for a work queue.

BC-12945 The web service API BillingAPI.previewInstallmentsPlanInvoices can now return invoice previews for currencies that are not the currency of the MainAccount of an AccountCurrencyGroup.

BC-12975	<p>Internal BillingCenter code included a validation that required that a producer write-off have an amount no greater than the ProducerWriteoffThreshold value from the producer's agency bill plan.</p> <p>This validation was redundant because identical configurable validation existed in the agency bill payment wizard user interface and in the agency payment API. The internal BillingCenter validation code was removed because it was not configurable.</p> <p>Note: If you have added configuration code that executes agency bill payments with write-off amounts directly in your configuration code, without going through the BillingCenter user interface or API that has configurable validation that checks the ProducerWriteoffThreshold, you must update your configuration to add this validation to your configuration code.</p>
BC-12996	Improved performance of the Automatic Disbursement batch process by adding a covering index.
BC-13175	<p>Made an API modification to the DirectBillPayment plugin. The function addInvoiceItemsDistributionCriteriaForUnappliedFunds now takes a ZeroDollarDMR parameter and not an UnappliedFund. This change guarantees that the payment allocation plan always has access to a non-null DirectBillMoneyReceived object during the allocation process, which is necessary for some allocation methods, such as Up To Amount of Next Invoice, to work correctly.</p> <p>In the base configuration, this change has no impact. The DirectBillPaymentFactory class was changed so that the ZeroDollarDMR object would be created before looking for eligible items, and then thrown away if no payment or allocation occurred. Do the same thing in your code. When using this method, create a new ZeroDollarDMR object, and then throw it away if it is unused.</p>

Other – Persistence

PL-25820	Fixed an issue with whitespace not being trimmed by trimming unicode full-width whitespace.
----------	---

PCIN – PolicyCenter Integration

BC-11302	<p>A Currency field has been added to the IssuePolicyInfo entity. This entity is passed in the issuePolicyInfo parameter to the web service method BillingAPI.issuePolicyPeriod.</p> <p>A call to BillingAPI.issuePolicyPeriod can now issue a policy to the sibling account with the correct currency.</p>
BC-11303	The web service method BillingAPI.searchForAccounts can now search for ListBill accounts that use a specified currency. The currency is ignored for accounts that are not of type ListBill. This search is implemented in the class BCAccountSearchCriteria, which is the type of the searchCriteria parameter of the searchForAccounts method.
BC-11304	The BillingAPI.createAccount web service method now accepts a parameter of type Currency to support creating an account in a currency other than the system's default currency.
BC-11384	The web service method BillingAPI.updateAccount now updates both the main account and its sibling accounts.
BC-11385	<p>A Currency field has been added to the plan info objects returned by the following methods of the BillingAPI web service:</p> <ul style="list-style-type: none"> • getAgencyBillPlans • getAllCommissionPlans • getAllPaymentPlans
BC-11386	The search algorithm for the web service method BillingAPI.getAllSubAccounts has been updated. This method now searches the subaccounts associated with account currency groups. If an account currency group contains accounts that have subaccounts, the method returns the main account of that account currency group.
BC-11387	<p>A Currency parameter has been added to the web service method BillingAPI.getPaymentPlansForAccount. This method now returns all payment plans in the specified currency for the account.</p> <ul style="list-style-type: none"> • If the Currency parameter is null, getPaymentPlansForAccount returns all the payment plans for the account. • For a ListBill account, getPaymentPlansForAccount returns an error if the Currency parameter does not match the currency on the account.
BC-11388	<p>An optional Currency parameter has been added to the web service method BillingAPI.getAccountInvoiceStreams.</p> <ul style="list-style-type: none"> • If the Currency parameter is not specified, the method returns all the invoice streams for the account. • If the Currency parameter is specified, the method returns all the invoice streams in the specified currency for the account.

BC-11389	<p>The web service method <code>BillingAPI.createProducer</code> has been updated to support a producer that conducts business in multiple currencies.</p> <p>This method now creates a group of producers in a <code>ProducerCurrencyGroup</code>, with one main producer associated with the preferred currency and sibling producers associated with the other supported currencies.</p>
BC-12782	<p>There was a problem with the unapplied fund amount being returned by the web service method <code>BillingSummaryAPI.retrieveAccountBillingSummary</code>. The unapplied fund amount included only the default unapplied fund and not the total for all unapplied funds for the account. This problem has been fixed, and the method now returns the total for all unapplied funds for the account.</p>
BC-12861	<p>The <code>BillingAPI</code> web service has special behavior for policy changes with a modification date that is the same as the policy period's effective date. There was a problem with policy changes that had a modification date that was not exactly at midnight, but was still on the same day as the policy period's effective date. In that case the webservice was incorrectly treating the policy change as if the modification date was not the same as the policy period's effective date. This problem has been fixed.</p>
BC-12919	<p>The web service method <code>BillingAPI.createProducerCode</code> now accepts an array of <code>CommissionPlanID</code> objects and creates matching producer codes for different currencies.</p>
BC-12941	<p>A call to <code>BillingAPI.updateProducerCode</code> now updates the producer code for all sibling producers.</p>
BC-12942	<p>A call to <code>BillingAPI.getAvailableBillingMethods</code> now returns all the billing methods that the producer supports for the given currency.</p>
BC-13036	<p>A call to <code>BillingSummaryAPI.retrieveAccountBillingSummary</code> now returns an array containing separate entries for each account in a mixed currency account.</p>
BC-13037	<p>If the overriding payer is a sibling account, a call to <code>BillingSummaryAPI.retrievePolicyBillingSummary</code> returns the main account as the overriding payer.</p>
BC-13038	<p>A call to <code>BillingSummaryAPI.retrievePeriodsBilledToAccount</code> now returns the policy periods (paid not owned) for all the sibling accounts of the main account.</p>
BC-13039	<p>A call to <code>BillingSummaryAPI.retrievePeriodsForAccount</code> now returns all the policy periods for both the main account and its sibling accounts.</p>

PCIN – PolicyCenter Integration, CONT – Contacts

BC-13269	<p>Instead of returning <code>null</code>, <code>ContactEnhancement.syncWithContactManager</code> now returns the locally existing version of the contact if it could not be found in <code>ContactManager</code>.</p>
BC-12918	<p>The web service API <code>BillingAPI.updateProducer</code> now updates information on any sibling producers in the <code>ProducerCurrencyGroup</code>.</p> <p>Additionally, when <code>updateProducer</code> includes a change of primary contact, <code>producer.PrimaryContact</code> is propagated to all sibling producers.</p> <p>Note: When a new primary contact is specified in the <code>BillingAPI.updateProducer</code> API call, the API call creates a new contact, marks it as Primary, and marks the existing one as Secondary. However, this API does not support modifying details of the existing primary contact, even if you provide a matching <code>AddressBookUID</code> for that contact. You must call <code>BillingAPI.updateContact</code> if you want to update details of the current primary contact.</p>
PCIN – PolicyCenter Integration, I18N - Internationalization	

PCIN – PolicyCenter Integration, I18N - Internationalization

BC-12691	<p>There was a problem with Account contact names sent from PolicyCenter that had a particle, such as <i>de la</i> in the French name Ami de la Torre. The particle was not included in the Account contact name in BillingCenter. This problem has been fixed, and now BillingCenter includes the particle portion of contact names.</p>
----------	---

PERF – Performance (INTERNAL)

BC-11688	<p>Added an index to the <code>PolicyPeriod</code> entity to reduce the time required by the <code>FullPayDiscount</code> work queue to find policy periods eligible for full pay discount.</p>
----------	---

PMDS – Payment Distribution

BC-13104	<p>Account negative invoice items are now also handled by <code>PaymentAllocationPlan</code>.</p>
----------	---

PMNT – Payment

BC-12871	<p>Targeting of unsettled invoice items is no longer configurable in the <code>IDirectBillPayment</code> plugin. This targeting is now handled implicitly and cannot be changed.</p>
----------	--

BC-12908	<p>There was a performance problem with the New Payment batch process, which queries for Direct Bill Money Received objects that do not have a corresponding <code>DirectBillPayment</code>. That query previously retrieved monies that were targeting accounts that had no money in the unapplied T-account. These monies would be picked up by the query and then discarded because there were no funds in the unapplied T-account to distribute. Now the query excludes any monies that are targeting accounts with a non-positive amount in their unapplied T-accounts, thus avoiding unnecessary processing by the batch process.</p>
BC-12927	<p>BillingCenter no longer creates invoice Items of type Down Payment Adjustment. These invoices used to be created during a payment plan change when:</p> <ul style="list-style-type: none"> • The new payment plan had a higher expected down payment amount. • The original down payment item was already billed. • The option Planned Only was selected. <p>In this scenario, BillingCenter created an invoice item of the difference. This invoice item had the type Down Payment Adjustment, and its event date was set to the original down payment's event date.</p> <p>Additionally, BillingCenter used to create invoice Items of type Down Payment Adjustment when installments were removed from the old schedule, but there was not a scheduled date in the new plan's date sequence that would fit in the remaining Policy Period's invoicing window.</p> <p>Now BillingCenter does not attempt to adjust the total down payment amount if the original down payment was not selected for modification. If the down payment is billed, but is then selected for modification by the payment plan change operation, it will be reversed and a new one created with the correct date. Otherwise, the down payment is left alone, and any new items will be of type installment. For the scenario in which there are no schedule dates remaining, the most recent bill date in the new date sequence is chosen for the new installment item.</p>
BC-12980	<p>The Direct Bill Payment screen has been split into three screens, Payments, Credit Distributions, and Payment Requests. For example, to access the Payments screen, navigate to Accounts tab → Payments → Payments. Each screen has a drop-down filter that can limit the database query to Last 30 Days, Last 60 Days, Last 90 Days, or All. The filters have different names on different screens: Payment Received, Credit Distribution Received, and Request Sent.</p>
BC-12981	<p>Removed the Recalculate button from the New Direct Bill Payment screen. All the fields that affect the distribution now post on change. To open this screen, you can navigate to Accounts tab → Actions → New Payment → New Direct Bill Payment.</p>
BC-13110	<p>Fixed an error that was occurring when a suspense item was edited and then released.</p>

PMPN – Payment Plan and Invoicing

BC-12350	<p>All payments and credit distributions on a policy charge with and without mixed items now redistribute after a payment plan change or an item amount change (only direct bill). This wasn't the case in previous releases.</p> <p>Redistribution after a payment plan change or item amount modification now uses a new strategy, <code>ProRataReversedPaymentAmountRedistributionStrategy</code>, which distributes the reversed amount only to same-sign items as the reversed amount on the charge. The strategy also distributes only up to the amount of the reversed amount, even when it is negative. The items to allocate are still picked by using pro rata distribution.</p>
BC-13010	<p>The performance of payment plan refactoring has been significantly improved.</p>
BC-13025	<p>A call to <code>BillingAPI.getPolicyPeriod</code> now returns the given policy period unless the requested period has an overriding payer that is a sibling account. In that case, the method returns the policy period for the main account.</p>
BC-13040	<p>A call to <code>BillingSummaryAPI.retrieveInvoicesForAccount</code> now returns all invoices for both the main account and its sibling accounts.</p>
BC-13052	<p>A call to <code>BillingAPI.getPaymentPlansFor</code> no longer returns ineligible <code>PaymentPlan</code> objects in the list of plans belonging to a <code>ListBill</code> account.</p>
BC-13053	<p>Added the following new item filters to the Change Payment Plan screen, available when you have a policy open:</p> <ul style="list-style-type: none"> • A new radio button, Not Fully Paid Items. Selecting this option includes any items that are not fully paid for payment plan change processing. • A new check box, Include Down Payment Items. Selecting this check box includes the down payment item for payment plan change processing. This checkbox is visible and available for editing only if Planned Items or Not Fully Paid Items is selected.

To open this screen in BillingCenter, with a policy open, navigate to **Payment Schedule → Edit Schedule**.

BC-13112	A new screen enables users to create and modify payment allocation plans. To open this screen, in BillingCenter, navigate to Administration → Business Settings → Payment Allocation Plans . These plans are used to determine which invoice items are paid as well as the order in which they are paid during an automatic direct bill payment.
PROD - Producer	
BC-13018	A call to <code>BillingAPI.createProducer</code> now requires a <code>PCNewProducerInfo</code> object, which is a subclass of <code>PCProducerInfo</code> that requires a <code>PreferredCurrency</code> . The <code>PreferredCurrency</code> specifies which currency the main producer in a <code>ProducerCurrencyGroup</code> will be created with. All other currencies will be set on sibling producers in the same <code>ProducerCurrencyGroup</code> .
Profiling	
PL-28172	Fixed an issue in which the Purge Profiler Data batch job would throw exceptions when purging web services profiling data.
Queries	
PL-18578	<p>This release adds a new class, <code>gw.api.database.QuerySelectColumns</code>, with static methods that help you specify the columns that you want selected in a row query. Instead of passing a Gosu block to the <code>select</code> method, you pass a list of <code>IQuerySelectColumn</code> objects, which you construct by using static methods on the <code>QuerySelectColumns</code> class. Each <code>IQuerySelectColumn</code> object represents a column in the result.</p> <p>For example, the following Gosu sample code creates a row query, with <code>Address</code> as the primary entity. The result includes only the <code>City</code> column for <code>Address</code> instances that match the query criteria.</p> <pre>uses gw.api.database.Query uses gw.api.database.QuerySelectColumns uses gw.api.path.Paths var addressQuery = Query.make(Address) ... // Join and condition statements go here. ... var addressResult = addressQuery.select({QuerySelectColumns.path(Paths.make(Address#City))})</pre> <p>The <code>QuerySelectColumns</code> class includes static methods that represent database functions to help you construct aggregate queries. For example, the following Gosu code returns a count of all <code>Address</code> instances.</p> <pre>uses gw.api.database.DBFunction uses gw.api.database.Query uses gw.api.database.QuerySelectColumns uses gw.api.path.Paths var addressQuery = Query.make(Address). var addressQyquery.withDistinct(true) // Always run aggregate queries // with Distinct set to true. addressResult = addressQuery.select({QuerySelectColumns.dbFunction(DBFunction.Count(Paths.make(Address#City))) })</pre>
PL-27474	Fixed an issue in which the <code>Count</code> property on a <code>gw.api.database.Query</code> was producing an incorrect SQL statement and count result when the <code>withDistinct(true)</code> setting was combined with a <code>Join</code> .
RNWL – Renewal Process, REWR – Rewrite Process	
BC-12989	<p><code>BillingAPI</code> now supports changing currency on a request from PolicyCenter to renew or rewrite a policy period. A rewrite request now also uses a <code>PaymentPlanID</code> because payment plans can be changed. Plan selection also matches the new currency. The methods are:</p> <ul style="list-style-type: none"> • <code>BillingAPI.renewPolicyPeriod</code> • <code>BillingAPI.rewritePolicyPeriod</code>

SOAP – SOAP APIs

BC-12880	<p>The trouble ticket methods in the deprecated RPCE web service IBillingCenterAPI have been added to the WS-I web services BCAPI and TroubleTicketAPI. The converted trouble ticket methods use a data transfer object (DTO) that encapsulates entity data.</p> <p>The methods added to BCAPI in this fix are:</p> <ul style="list-style-type: none"> • addNoteToAccount • addNoteToPolicyPeriod • addDocumentToAccount • addDocumentToPolicyPeriod • addDocumentToProducer <p>In addition, the following methods that put holds and release holds on accounts, policy periods, and producers for trouble tickets have been added to the web service TroubleTicketAPI:</p> <ul style="list-style-type: none"> • putHoldOnAccount • putHoldOnPolicyPeriod • putHoldOnProducer • releaseHoldOnAccount • releaseHoldOnPolicyPeriod • releaseHoldOnProducer <p>The following TroubleTicketAPI method has been deprecated:</p> <ul style="list-style-type: none"> • putHoldOnPolicy
BC-12930	<p>The following methods have been added to the BCAPI web service:</p> <ul style="list-style-type: none"> • createSharedActivity provides the same functionality as the deprecated IBillingCenterAPI.createSharedActivity web service method. • createActivity creates an activity and uses auto-assignment to assign it to a user. • createActivityAssignedToUser creates an activity and assigns it to a user. <p>All three methods accept a trouble ticket data transfer object of type TroubleTicketDTO.</p>
BC-12969	<p>The getAccountInfo method has been added to the BCAPI web service. This method returns an AccountInfoDTO object.</p> <p>This method replaces the deprecated web service method IBillingCenterAPI.getAccountInfo.</p>
BC-12971	<p>RPCE web services have been deprecated. Use the new WSI web service implementations instead. The deprecated web services are:</p> <ul style="list-style-type: none"> • BCWebService.gs • IBillingCenterAPI.gs • ICollateralAPI.gs • IMaintenanceToolsAPI.gs • IMessagingToolsAPI.gs • InvoiceAPI.gs • IPolicyPeriodAPI.gs • IStagingTableEncryptionAPI.gs
BC-12995	<p>The new method changeBillingMethodToListBill has been added to the BCAPI web service. This method changes the billing method for a policy period from Agency Bill or Direct Bill to List Bill. The list bill account must be created prior to calling this method.</p>
BC-13043	<p>The following methods have been added to the BCAPI web service:</p> <ul style="list-style-type: none"> • startDelinquencyOnAccount – Does not check if the account already has an active delinquency process. • startDelinquencyOnPolicyPeriod – Does not check if the policy period already has an active delinquency process. • triggerDelinquencyWorkflowOnAccount – Triggers the workflow for the active delinquency processes on the account, including those of the policy periods of the account. • triggerDelinquencyWorkflowOnPolicyPeriod – Triggers the workflow for the single active delinquency process associated with the policy period. <p>These methods replace the following deprecated methods:</p> <ul style="list-style-type: none"> • IBillingCenterAPI.startDelinquency • IBillingCenterAPI.triggerDelinquencyWorkflow
BC-13122	<p>The following two methods have been added to the BCAPI web service:</p> <ul style="list-style-type: none"> • cancelPremiumReportDueDate provides the same functionality as the deprecated IBillingCenterAPI.cancelPremiumReportDueDate. • createPremiumReportDueDate is a new method that accepts a data transfer object of type PremiumReportDueDateDTO.

BC-13136	There is a new method, <code>BillingAPI.previewCommissionRates</code> , that enables retrieval of commission rate overrides for display in a policy system. This method replaces the deprecated <code>IBillingCenterAPI.populateCommissionRateOverrides</code> .
----------	--

SRCH – Searching

BC-11984	AccountSearchCriteria has been converted from Java to Gosu. As part of upgrade, if you had any extensions in your <code>AccountSearchCriteria.etc</code> file, you will have to code them into the new Gosu search criteria gosu file and remove <code>AccountSearchCriteria.etc</code> . See "Working with Search Criteria in Gosu" in the BillingCenter Configuration Guide for more information.
BC-11986	The following search criteria have been converted from Java to Gosu: <ul style="list-style-type: none">• DelinquencySearchCriteria• InvoiceSearchCriteria• PaymentSearchCriteria• ProducerSearchCriteria• TroubleTicketSearchCriteria• AgencyMoneyReceivedSearchCriteria If you added an extension to the original entity file, you must make the corresponding code changes to the Gosu search criteria file and then delete the extension file. See "Working with Search Criteria in Gosu" in the BillingCenter Configuration Guide for more information.

User Interface – Fit & Finish

BC-3697	There was a problem that occurred when moving between Step 1 and Step 2 of the wizards used to create a charge for a new policy or a policy change. Selecting a charge-level payer or invoice stream the second time was causing erroneous errors. This problem has been fixed, and charge-level overrides now persist when navigating back and forth in the wizard.
BC-12429	Fixed a problem in the screen for creating an invoice stream for an account, which was throwing a <code>NullPointerException</code> if an empty string was entered for the unapplied fund description.
BC-12571	The Events column on the Workflow card of the Delinquency Plan screen now shows more text.
BC-12573	The Invoice field on the Multiple Payment Entry screen is now validated when moving to the next screen, as are the other fields on this screen. You can access this screen by navigating to Desktop tab -> Actions -> New Payment -> Multiple Payment Entry .
BC-12741	The Sample Data screen, used to load BillingCenter sample data, has been renamed and moved. This screen is now located on the Internal Tools tab and is named BC Sample Data . To access this screen, press Alt+Shift+T . Then click the Internal Tools tab and click BC Sample Data in the sidebar.
BC-12790	There was a problem in the user interface that occurred after removing a row from the Return Premium Handling scheme. When removing a row made an existing row become the next-to-last row, the Move-Down button continued to be available for the row. Because the next-to-last row is the row above the default row, Other , it must not be moved down. The Other row must always be the last one. This problem has been fixed.
BC-13117	There was a problem in the user interface for creating invoice streams that has been fixed. BillingCenter no longer throws an exception when the user does the following: <ol style="list-style-type: none">1. Creates an unapplied fund for one stream.2. Configures another stream to use the same unapplied fund.3. Returns to the first stream and modifies the fund's description.

User Interface – Fit & Finish, AGBL – Agency Bill

BC-12306	Fixed a problem with list views that had a footer total. These list views were not able to sort by column. Now the user can sort by column in list views that have a footer total. For example, sorting by column works for the Agency Bill Open Items , Agency Bill Cycles , and Agency Bill Payments screens.
----------	--

BC-12335	<p>The Agency Bill Credit Distribution wizard's detail panel on Steps 2 and 3 has been changed to match the similar panels of the Agency Bill Payment Wizard. Changes include:</p> <ul style="list-style-type: none"> For both creating and editing, the field Remaining replaces the field Total under This Credit Distribution: <ul style="list-style-type: none"> For creating, Remaining = Credit - Distributed - In Suspense For editing, Remaining = Total Available - Distributed - In Suspense For creating, the Credit Distribution Info section remains the same by just displaying Credit, which is the value of Producer Unapplied. For editing, the fields under Credit Distribution Info have been modified to also display the fields Other Unapplied and Total Unapplied. In this section, the following fields have the following values: <ul style="list-style-type: none"> Credit - What was distributed on this credit distribution that you are about to edit. Other Unapplied – The value is Producer Unapplied. Total Available – The value is Credit + Other Unapplied.
BC-12705	<p>There was an issue with the Sort By drop-down list on the Agency Bill Cycles screen that is no longer an issue because this drop-down list has been removed from this screen. Sorting in this screen is now performed by clicking on the heading of the column you want to sort by.</p>
BC-12709	<p>New columns have been added for Policy Number and Insured Name to the Agency Bill Statement screen in PolicyActivityLV.itemLevel.pcf. The user can group the data by column, as needed. The sorting drop-down list has been removed because sorting can now be performed by clicking individual column headers.</p>
BC-12952	<p>Nested iterators have been removed from the following screens and the data has been flattened to enable sorting by column:</p> <ul style="list-style-type: none"> Agency Bill Cycles (PolicyActivityLV.itemlevel) Agency Bill Open Items (PolicyActivityLV.itemlevel) Agency Bill Payments (AgencyDistItemsReadonlyCV) Agency Bill Promises (AgencyDistItemsReadonlyCV)
BC-12986	<p>Added two new fields to the Agency Bill Policy Details popup:</p> <ul style="list-style-type: none"> Account Number Account Name <p>The fields are in the Account Information section of this popup.</p> <p>To open the popup, open a producer and navigate to Agency Bill Payments → Executed Payments. Select a payment transaction and click Edit to open the Agency Bill Payments screen, where you can start the Agency Payment wizard. If you select a distribution of Statement and Policies, you can access the popup by clicking a statement number and then clicking the associated policy number.</p>

User Interface – Fit & Finish, TTKT – Trouble Ticket

BC-11744	The second step of the New Trouble Ticket wizard now has a Name column instead of an Entity ID column.
----------	---

Web – ListViews

PL-10908	Added the groupedOnEnter attribute to many cell-based PCF elements. If true, the ListView is grouped by this cell upon entering the page. Only one grouped cell is allowed at any given time, and it is applicable only when the column is sortable.
----------	--

PL-27732	A new height attribute has been added to the ListViewPanel PCF element. This sets the vertical size, in pixels, of the list view. If the list data is taller than the specified height, then a vertical scroll bar appears within the list view. The height is calculated from the list view header toolbar. Header rows are fixed; the footer scrolls with the data. Note that this is currently an experimental attribute and may not function properly. You should fully test any use of this attribute.
----------	---

PL-28466	Fixed an issue with column reordering when switching between filtering views in a list view.
----------	--

Web – UI/Runtime

PL-28749	Fixed a critical security vulnerability to persistent cross-site scripting attacks.
----------	---

PL-27884	The following new methods were added to the Javascript gw.api.Util class: getValue , setValue , getValues , setValues . These methods enable you to get or set values of input elements. For example, you can use these methods in TemplatePanel and pass Gosu variables as arguments into these methods.
----------	---

PL-28336	Fixed an issue that would occur when there was an action defined on a range cell.
----------	---

Work Queues

PL-27301	Improved the server performance when selecting available work queue items.
----------	--

PL-28696 Fixed an issue with workItem orphan detection during daylight saving time adjustment.

XMLElement (and XSD types)

PL-27131 Gosu XML/XSD types now support use of circular xs:include references.

Known Issues and Limitations

This topic describes known issues with this release of Guidewire BillingCenter.

Note: For maintenance releases, Guidewire often defers fixing configuration issues that require merging files during the upgrade. Workarounds to many of these issues are listed in the following topics. The goal of this policy is to make upgrades as straightforward as possible.

This topic includes:

- BillingCenter Known Issues
- Studio/Platform Issues

BillingCenter Known Issues

Rewrite Billing Instruction with the same policy number as the original policy is not incrementing the TermNumber and throws DBDuplicateKeyException (BC-2961)

Issue – If term number "1" already exists for the policy, sending a Rewrite billing instruction without specifying the term number will throw a DBDuplicateKeyException because BillingCenter uses the number "1" for term number.

Workaround – Include a term number that is unique for the given policy number when sending a rewrite billing instruction.

Total Payment Applied on agency bill statement wrong after payment plan edit (BC-2966)

Issue – The Total Payment Applied amount will be incorrect on the next agency bill statement if the following are also true:

- Producer has at least two policies
- Previous agency bill statement is billed and partially paid
- Payment plan used by one of the policies is modified

Workaround – Guidewire is aware of this issue.

Reversing write-off commissions and write-off commission recovery not in the user interface (BC-7184)

Issue – The BillingCenter user interface does not provide a way to reverse write-off commissions or write-off commission recovery.

Workaround – Clone the Reverse button under the Payments tab and reverse CmsnPayableReduction and other transactions using the following action: CurrentLocation.startEditing(), producerTransaction.reverse(), and CurrentLocation.commit().

Commission: Commission override after a commission write-off causes write-off to be reserved (BC-9536)

Issue – Several methods of the BillingAPI WS-I webservice, such as `issuePolicyPeriod` and `cancelPolicyperiod`, indirectly call `BillingInstructionUtil.executeAndCommit`, which commits the current bundle. These methods also directly call `tryCatch`, which commits a new bundle after executing the entire block of code inside `tryCatch`. As a result, the code tries to commit the same bundle twice.

Following are two ways you can encounter this problem:

Case 1

If you try to create an activity in the block of code passed to `tryCatch` after the call to execute the billing instruction, BillingCenter throws `DBA1readyExecutedException`.

For example:

```
function issuePolicyPeriod(issuePolicyInfo : IssuePolicyInfo, transactionId : String) : String{
    require(issuePolicyInfo, "issuePolicyInfo")
    var publicID = tryCatch( \ bundle -> {
        var pid= issuePolicyInfo.executeIssuanceBI()
        // code *here* to create a new activity
        return pid
    }, transactionId)
    return publicID
}
```

Case 2

If you used `CreateTroubleTicketHelper` to create a trouble ticket in a billing instruction preupdate rule, you would get a `DBA1readyExecutedException`. The following is an example of a billing instruction preupdate rule that would trigger the error:

```
// create trouble ticket with a delq hold until second invoice
var troubleTicketHelper = new CreateTroubleTicketHelper()
var ticket = troubleTicketHelper.createTroubleTicket(
    billingInstruction as PlcyBillingInstruction).PolicyPeriod)
ticket.Title = "Delinquency hold on issuance"
ticket.DetailedDescription = ticket.Title
ticket.Priority = "normal"
ticket.TicketType = "automatic"
ticket.Hold.setAppliedToHoldType(HoldType.TC_DELINQUENCY, true)
ticket.Hold.setReleaseDate(HoldType.TC_DELINQUENCY, gw.util.GWBaseDateEnhancement.CurrentDate)
```

Workarounds

- For Case 1, in the block of code passed to `tryCatch`, do not create or update entities after the call to execute the billing instruction. For example:

```
function issuePolicyPeriod(issuePolicyInfo : IssuePolicyInfo, transactionId : String) : String{
    require(issuePolicyInfo, "issuePolicyInfo")
    var publicID = tryCatch( \ bundle -> {
        // code *here* to create a new activity
        var pid= issuePolicyInfo.executeIssuanceBI()
        return pid
    }, transactionId)
    return publicID
}
```

- For Case 2, instead of using `CreateTroubleTicketHelper` to create the trouble ticket, create related entities directly as in the following example in a PolicyPeriod enhancement:

```
var ticket = new TroubleTicket(this)
var ttJoinEntity = new TroubleTicketJoinEntity(ticket)
ticket.Title = ticketTitle
ticket.DetailedDescription = "Delinquency hold placed on Account with a " + ticket.Title
ticket.Priority = "normal"
ticket.TicketType = ticketType
ticket.Hold.setAppliedToHoldType( HoldType.TC_DELINQUENCY, true )
ticket.Hold.setReleaseDate(HoldType.TC_DELINQUENCY, releaseDate)
ticket.addToTroubleTicketJoinEntities(ttJoinEntity)
this.addToTroubleTicketJoinEntities(ttJoinEntity)
this.Account.addToTroubleTicketJoinEntities(ttJoinEntity)
ticket.Hold.checkForHoldAdditions()
```

PolicyCenter integration on renewal offer: suspense payment is not being applied to policy period (BC-9691)

Issue – When BillingCenter is integrated with PolicyCenter, a suspense payment that is created with an offer number from a PolicyCenter renewal is incorrectly applied to the owner account of the policy period. Correct behavior would be to apply the suspense payment to the renewal policy period instead of the owner account.

Workaround – Guidewire is aware of this issue.

Payment Distribution: BillingCenter incorrectly distributes payment amounts that have been disbursed (BC-9783)

Issue – Reversing a DirectBillPayment is causing T-account unapplied to become negative. This can happen when another direct bill payment on the same account gets modified.

Workaround – Create a recapture charge when unapplied goes below zero.

Net Owed Amount calculation for promises is sometimes incorrect (BC-9920)

Issue – The Net Owed amount is calculated incorrectly for invoice items that have been fully or partially promised.

Workaround – Guidewire is aware of this issue.

Saved promises are listed in two places after upgrade (BC-10019)

Issue – After upgrade, saved promises are also listed under Executed Promises location group.

Workaround – Execute saved promises prior to upgrade.

Inconsistency in behavior when an invoice item (with item level commission rate override) is removed from a charge (BC-10368)

Issue – When an invoice item with item-level commission rate override is modified, BillingCenter recalculates commission inconsistently depending on how the invoice item is changed.

For example, you can modify an existing invoice item on a charge. Alternatively, you can add a new invoice item to the charge and delete the existing invoice item. BillingCenter will calculate the resulting commission differently for these two cases.

Workaround – Guidewire is aware of this issue.

AccountImpl.getNotFullyAllottedFundsTrackers is not bundle safe (BC-11558)

Issue – Manipulating bundles for funds tracking should be avoided. All linked entities are not automatically removed when you remove entities from the bundle.

Workaround – Guidewire is aware of this issue.

Localization - Some labels on new document creation for account/ producer not localized (BC-12136)

Issue – Some labels are not translated into the language the user has selected when the user adds a new document.

Workaround – Guidewire is aware of this issue.

[Payment redistribution behavior in payment plan refactor changed \(BC-12416\)](#)

Issue – In BillingCenter 7, payment plan refactor redistributed the entire payment. This behavior has changed as follows:

- In BillingCenter 8, payment plan refactor redistributes only the amount of the payment that had been originally allocated.
- In BillingCenter 7, payment plan refactor respects the account's distribution method. In BillingCenter 8, it does not because an account has no distribution method.

Workaround – If you would like to maintain BillingCenter 7 behavior, contact Guidewire BillingCenter Product Management.

[Integrating PolicyCenter and BillingCenter with multicurrency might require additional configuration \(BC-12856\)](#)

Issue – The base configurations of PolicyCenter 8.0 and BillingCenter 8.0 include an integration that permits single-currency interoperation of the two applications. To integrate PolicyCenter 8.0 and BillingCenter 8.0 in a multicurrency environment, you might need to extend the default implementation.

Workaround – Guidewire is aware of this issue.

[Upgrading from BillingCenter 3.0.5 or earlier to 8.0 can result in policy commission errors \(BC-13101\)](#)

Issue – In BillingCenter 3.0.5 and earlier, transferring a policy from one producer to a different one and then back to the original corrupted the data by creating duplicate policy commissions. Part of the fix involved an upgrade trigger that consolidated these policy commissions, but this upgrade did not redistribute the commission correctly to the individual invoice items afterward.

Workaround – If you are upgrading from BillingCenter 3.0.5 or earlier, before upgrading to BillingCenter 8.0, first do the minor upgrade to BillingCenter 3.0.6 or a later 3.0 release. After you do the upgrade to BillingCenter 8.0, contact Customer Support for SQL statements and Gosu code that fix any incorrect invoice items resulting from incorrectly redistributed commissions.

[Payment reversal does not trigger delinquency \(BC-13286\)](#)

Issue – When a payment against a past due item is reversed, BillingCenter does not trigger delinquency.

More specifically, starting with release 8.0.1, BillingCenter triggers delinquency only from the `InvoiceDue` batch process. Doing so enables you to avoid delinquency on payment reversal, for example, by applying credit items or some other changes. The problem is that if an invoice has already been marked Due, the `InvoiceDue` batch process does not process the invoice again, and does not trigger delinquency.

Workaround – Guidewire is aware of this issue.

[Installments are not numbered correctly after a payment plan change \(BC-13343\)](#)

Issue – After a payment plan change, new installments that are created start at a value that is not consecutive with installments retained from the previous plan. If all installments from the previous plan are removed, the new installments incorrectly start with a value greater than 1.

When you refactor, you can choose to refactor all invoices or a subset of the invoices. So, it is possible that all installments from the old plan can be removed, or that some can be retained. If all old installments are removed, the new installments are supposed to start with the number 1. If some installments are retained, the new installments are supposed to start with a value that is consecutively after the old installments. This renumbering is not being done correctly.

Workaround – Guidewire is aware of this issue.

On renewal, cannot remove an alternate payer (BC-13390)

Issue – BillingCenter fails to process the removal of an overriding payer correctly when the request is included in the same billing instruction with a policy renewal request.

For example, if a list bill policy is converted to direct bill on renewal, BillingCenter does not remove the overriding payer from the renewed policy period. The renewal may fail, or it may succeed but incorrectly leave the renewal policy period with an overriding payer.

Workaround – Contact Guidewire Customer Support for assistance.

Studio/Platform Issues

Problem with regen-java-api command and JAR files (PL-16351)

Issue – If you run the gwbc regen-java-api command, BillingCenter creates a *BillingCenter/java-api/lib* directory and puts JAR files for Java development in this directory. However, BillingCenter does not always perform this task correctly, resulting in some of the generated JAR files not being copied to this directory.

Workaround – For missing library files, copy or import into your Java development environment the missing library files from the following directory:

BillingCenter/admin/lib

Copy them into the following directory:

BillingCenter/java-api/lib

Renaming method or property throws ParseResultsException (PL-16633)

Issue – If you rename a property or a method or you change a method signature, and a workflow references that property or method in a Gosu field, BillingCenter throws `ParseResultsException`. This is the intended behavior.

Workaround – Restart the workflow engine. To do so:

1. Log in to BillingCenter using an administrative account.
2. Access Internal Tools → Reload.
3. Click Reload Workflow Engine.

Javadoc command does not generate index file in expected location (PL-27679)

Issue – When you run gwbc regen-java-api, an `index.html` file is not created in *BillingCenter/java-api/doc*.

Workaround – The command now generates Javadoc JAR files in *BillingCenter/java-api/doc*. To view the Javadoc, add the JAR files to the Studio project.

Gosu does not automatically downcast if the left side of the typeis or typeof expression uses deprecated members (PL-27724)

Issue – To improve readability of your Gosu code, Gosu automatically downcasts after a `typeis` expression if the type is a subtype of the original type. This is particularly valuable for `if` statements and similar Gosu structures. For example, if a variable has type `Object`, you can use code such as:

```
if( x typeis String ) {  
    length // NOTE: length is a property on String, but *not* on Object.  
}
```

In this release, Gosu does not automatically downcast if the left side of the `typeis` or `typeof` expression uses deprecated members. This may result in new compilation errors.

Workaround – To fix these compiler errors, explicitly downcast with the as keyword before you access properties or methods on the subtype but not on the original type. For example, suppose a property called Dep is deprecated:

```
if (x.Dep typeis ExampleType) {  
    return (x.Dep as ExampleType).PropertyOnExampleSubtype  
}
```

[Long text in table cells can add white space to the right of the page \(PL-28288\)](#)

Issue – In an editable list view, extremely long text entered in a single cell can cause additional white space on the right side of the page. Long text is text that occupies approximately the entire width of the screen. This issue occurs primarily in Chrome.

Workaround – If you expect users to enter large amounts of text into the cells of a column, configure the column to support text wrapping.

[AddressZoneValue in zone-config.xml files uses display name instead of code \(PL-28511\)](#)

Issue – Some of the <AddressZoneValue> expressions in the US, AU, CA, and DE zone-config.xml files use the DisplayName, rather than the Code, of the associated State.Abbreviation. Configuring BillingCenter to use translated names for the StateAbbreviation typelist can result in failure to find some zones for that country.

Workaround – Replace references to State.Abbreviation.DisplayName with State.Abbreviation.Code in <AddressZoneValue> elements.

In each zone-config.xml file, replace the following DisplayName property:

```
<AddressZoneValue>Address.State.Abbreviation.DisplayName</AddressZoneValue>
```

Instead, use the following Code property:

```
<AddressZoneValue>Address.State.Abbreviation.Code</AddressZoneValue>
```

[Security vulnerability – reflected XSS \(PL-29052\)](#)

Issue – There is a non-persistent cross-site scripting (reflected XSS) vulnerability. Unlike other XSS types, this vulnerability does not permit privilege escalation and does not propagate easily to other users.

Workaround – Guidewire is aware of the issue. Strong email filtering with phishing/malware detection is an effective defense against this type of attack. Contact Customer Support for more information.

[Problem with running regen-java-api command with deprecated=true flag \(PL-28992\)](#)

Issue – If you run the gwbc regen-java-api command with the flag -Ddeprecated=true, the command can fail.

Workaround – Specify dev-deploy as the value of the depends attribute, as follows:

1. Open the build.xml file in *BillingCenter/modules/ant* in an editor.

2. Find the entry that starts as follows:

```
<target name="regen-java-api" depends="init"
```

3. Change the value of the depends attribute to dev-deploy, as follows:

```
<target name="regen-java-api" depends="dev-deploy"
```

4. Save the file and then run the regen-java-api command.

Administrative command-line tools cannot refresh WSDL (PL-29021)

Issue – Some administrative command-line tools rely on web service implementation classes such as `MaintenanceToolsAPI.gs`. Source files for these classes use the `@Export` annotation, which allows you to edit the file. In this release, the administrative command-line tools cannot refresh the WSDL for these classes. Any change to the web service implementation class that changes the WSDL can prevent the administrative tools from working. Therefore, the only changes you can make to these classes are changes that do not affect the WSDL. For example, you can add `@WsPermission` annotations to change the permissions without changing the WSDL.

Workaround – Guidewire is aware of this issue.

Chrome browser cannot display product documentation in HTML format (DOC-7251)

Issue – If you use the Google Chrome browser, you can view the HTML Guidewire product documentation only if it is served by an HTTP server using the `http://` protocol. The Chrome browser cannot load HTML product documentation from your local disk by using the `file://` protocol.

Workaround – Use a different browser, such as Microsoft Internet Explorer. If you use the HTML documentation only to access the PDF files, you can open them directly in the `pdf` subdirectory of the `doc` directory.

New mechanism for reloading Gosu classes (DOC-8218)

Issue – In past releases, you could modify your PCF files and Gosu classes in Studio, and then reload the changes into the running server by pressing `Alt+Shift+L` in the application user interface. This shortcut no longer loads Gosu classes.

Workaround – To have the server reload your Gosu classes, in Studio, click **Build** → **Make Project**. When Studio is finished compiling your project, the changes will be loaded. You can also restart your server to load the Gosu classes.



chapter 20

Guidewire BillingCenter 8.0.0 Release Notes

Release 8.0.0

Document Revision: 02-July-2015

Copyright © 2001-2015 Guidewire Software, Inc. All rights reserved.

Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire Reinsurance Management, Guidewire ContactManager, Guidewire Vendor Data Management, Guidewire Client Data Management, Guidewire Rating Management, Guidewire InsuranceSuite, Guidewire ContactCenter, Guidewire Studio, Guidewire Product Designer, Guidewire Live, Guidewire DataHub, Guidewire InfoCenter, Guidewire Standard Reporting, Guidewire ExampleCenter, Guidewire Account Manager Portal, Guidewire Claim Portal, Guidewire Policyholder Portal, Gosu, Deliver Insurance Your Way, and the Guidewire logo are trademarks, service marks, or registered trademarks of Guidewire Software, Inc. in the United States and/or other countries.

All other trademarks are the property of their respective owners.

This material is confidential and proprietary to Guidewire and subject to the confidentiality terms in the applicable license agreement and/or separate nondisclosure agreement.

Guidewire products are protected by one or more United States patents.

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Support
- Issues and Major Changes
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire BillingCenter.

IMPORTANT If you skipped one or more upgrade releases to BillingCenter, be sure to read the Release Notes for those releases to learn about changes and fixed issues.

Version Number

This release of Guidewire BillingCenter is 8.0.0.

- If you are integrating with Guidewire PolicyCenter, this version of BillingCenter requires PolicyCenter 7.0.6 or later.
- If you are integrating with Guidewire ContactManager, this version of BillingCenter requires ContactManager 7.0.0.

Installing This Release

Refer to the following:

- The *BillingCenter Installation Guide* for general installation information
- Prior BillingCenter release notes for any versions that you have skipped

Support

For assistance, visit the Guidewire Resource Portal – <http://guidewire.custhelp.com>

Issues and Major Changes

This section contains issues and major changes that may affect your installation.

- Upgrade Support
- Base PCF File Changes
- Changes in This Release Provided in Upgrade Diff Report

Upgrade Support

Upgrading from BillingCenter 3.0.4 or newer versions is supported in this release.

The following information will be added to the *Upgrade Guide* in future releases:

[Upgrading from BillingCenter 2.1 to BillingCenter 8 requires upgrading to BillingCenter 7 first](#)

BillingCenter 8 supports direct upgrade only from BillingCenter 3.

To upgrade to BillingCenter 8 from BillingCenter 2.1, you must:

1. Upgrade to BillingCenter 7.

2. Run the following work queues:

```
com.guidewire.bc.domain.commission.impl.UpgradeCommissionWorkQueue  
com.guidewire.bc.domain.accounting.billinginstruction.impl.UpgradeBillingInstructionPayme  
ntPlanWorkQueue
```

3. Upgrade to BillingCenter 8.

Upgrade does not convert Money properties to MonetaryAmount properties in entity extensions

BillingCenter upgrade cannot automatically convert the `Money` data type to the `MonetaryAmount` data type. If you created entity extensions, the upgrade process will not upgrade your extensions that include properties that use the `Money` data type.

Before you upgrade, you can manually update any extension properties that use the `Money` data type.

If you change the definition of your extension properties to use the `MonetaryAmount` datatype rather than the `Money` data type, upgrade will correctly convert your entity extension data. Define the `MonetaryAmount` property as follows:

- The name of the new `MonetaryAmount` property is the same as the name of the `Money` property
- If the old `Money` property had a `columnName` attribute defined as something other than the `Money` property name, use that old `Money.columnName` as the name of the new `MonetaryAmount.amountColumnName` attribute.
- For extensions to entities that will be `InCurrencySilo` entities in BillingCenter 8, you must add a `DefaultValueZero` tag if your old `Money` property had the `default` attribute set to 0.
- Set `scaleToCurrency` to `true` unless you have a requirement to do otherwise.
- Set the `soapNullOk` attribute to `true`

The following examples show how you must redefine `Money` properties in your extensions to `MonetaryAmount` properties before you proceed with upgrade:

Old Total:

```
<column  
    name="Total"  
    type="money"/>
```

New Total:

```
<monetaryamount  
    name="Total"  
    amountColumnName="Total"/>
```

Old Total with default = 0:

```
<column  
    name="Total"  
    default="0"  
    type="money"/>
```

New Total:

```
<monetaryamount  
    name="Total"  
    amountColumnName="Total">  
    <tag name="DefaultValueZero"/>  
</monetaryamount>
```

Old Total where name and columnName differ:

```
<column  
    name=" Total"  
    columnName="totalColumn"  
    type="money"/>
```

New Total:

```
<monetaryamount  
    name="Total"  
    amountColumnName="totalColumn"/>
```

After you complete this configuration step, your database will upgrade successfully.

Upgrade fails if BCContact and PaymentDetails have customer extensions

If you are upgrading from BillingCenter 3 or earlier, and you have added extension fields added to the BCContact or PaymentDetails entities, upgrade does not migrate data in those fields to BillingCenter 8.

Before you upgrade, remove any extension fields from BCContact.etx and PaymentDetails.etx and add those extension fields to the corresponding supported entity. The extension field names and definitions must not change.

- Move any extension fields from BCContact.etx to Contact.etx.
- Move any extension fields from PaymentDetails.etx to PaymentReceipt.etx.

Note: You must not change the name or definition for each field you migrate from the old entity extension to the new entity extension.

Upgrade will not remove data rows for entities that have changed from retireable to non-retireable

In cases where the definition of an entity has been changed from being retireable to being non-retireable, BillingCenter 7.0.3 and later versions will not automatically remove data rows that had been retired. Instead, before upgrading to BillingCenter 8, a new version check,

MigrateTableFromRetireableToEditableVersionCheck, will run. This version check will log an error if any retired rows are found. If any such errors are logged, the retired rows must be examined and removed or restored before upgrade can continue. Contact Guidewire Customer Support for assistance.

Base PCF File Changes

The link below requires that the ReleaseNotes_files directory be on your local disk in the same directory as this release notes file.

BillingCenter release 7.0.4 to 8.0.0

To view a report of the changes to the base PCF files, [click here](#).

Changes in This Release Provided in Upgrade Diff Report

Guidewire provides a report detailing certain differences between the current release and your prior release. This report describes changes in display keys, entities, typelists, and the Gosu API. To obtain your custom Upgrade Diff Report, visit the Guidewire Resource Portal.

Known Issues and Limitations

This section describes known issues with this release of Guidewire BillingCenter.

- BillingCenter Known Issues
- Studio/Platform Issues

BillingCenter Known Issues

Integrating PolicyCenter and BillingCenter with multicurrency may require additional configuration (BC-12856)

Issue – The base configurations of PolicyCenter 8.0.0 and BillingCenter 8.0.0 include an integration that permits single-currency interoperation of the two applications. To integrate PolicyCenter 8.0.0 and BillingCenter 8.0.0 in a multicurrency environment, you may need to extend the default implementation.

Workaround – Guidewire is aware of this issue.

Contact Manager integration: BillingCenter creates two records for the same contact (BC-12752)

Issue – BillingCenter creates two records for the same contact if both BillingCenter and PolicyCenter have separately added the contact to their respective databases.

Workaround – The contacts must be manually updated.

Payment plan refactor error when trying to refactor an invoice item with a commission rate override (BC-12675)

Issue – Payment plan changes fail if any of the invoice items being updated have commission rate overrides.

Workaround – Guidewire is aware of this issue.

Upgrade fails when there is exactly one disbursement with an outgoing payment issued in the pre-upgrade database (BC-12851)

Issue – Upgrade fails when the entire application has sent exactly one disbursement.

Workaround – Make an additional disbursement to enable upgrade to succeed.

Payment redistribution behavior in payment plan refactor changed (BC-12416)

Issue – In BillingCenter 7, payment plan refactor redistributed the entire payment. This behavior has changed as follows:

- In BillingCenter 8, payment plan refactor redistributes only the amount of the payment that had been originally allocated.
- In BillingCenter 7, payment plan refactor respects the account's distribution method. In BillingCenter 8, it does not.

Workaround – If you would like to maintain BillingCenter 7 behavior, contact Guidewire BillingCenter Product Management.

Inconsistency in behavior when an invoice item (with item level commission rate override) is removed from a charge (BC-10368)

Issue – When an invoice item with item-level commission rate override is modified, BillingCenter recalculates commission inconsistently depending on how the invoice item is changed.

For example, you can modify an existing invoice item on a charge. Alternatively, you can add a new invoice item to the charge and delete the existing invoice item. BillingCenter will calculate the resulting commission differently for these two cases.

Workaround – Guidewire is aware of this issue.

Net Owed Amount calculation for promises is sometimes incorrect (BC-9920)

Issue – The Net Owed amount is calculated incorrectly for invoice items that have been fully or partially promised.

Workaround – Guidewire is aware of this issue.

Saved promises are listed in two places after upgrade (BC-10019)

Issue – After upgrade, saved promises are also listed under Executed Promises location group.

Workaround – Execute saved promises prior to upgrade.

Localization - Some labels on new document creation for account/ producer not localized (BC-12136)

Issue – Some labels are not translated into the language the user has selected when the user adds a new document.

Workaround – Guidewire is aware of this issue.

AccountImpl.getNotFullyAllottedFundsTrackers is not bundle safe (BC-11558)

Issue – Manipulating bundles for funds tracking should be avoided. All linked entities are not automatically removed when you remove entities from the bundle.

Workaround – Guidewire is aware of this issue.

Total Payment Applied on agency bill statement wrong after payment plan edit (BC-2966)

Issue – The Total Payment Applied amount will be incorrect on the next agency bill statement if the following are also true:

- Producer has at least two policies
- Previous agency bill statement is billed and partially paid
- Payment plan used by one of the policies is modified

Workaround – Guidewire is aware of this issue.

Agency Bill Payment wizard fails when attempting to execute a promise with no associated dist items (BC-4348)

Issue – If you click Execute on the Execute Agency Promise As Payment screen to apply a payment to a promise that has been pre-filled with zeroes, the following exception occurs:

EvaluationException: Cannot re-add AgencyCyclePayment since it has already been removed

Workaround – Use the Agency Bill Payment wizard to apply the payment instead.

Rewrite Billing Instruction with the same policy number as the original policy is not incrementing the TermNumber and throws DBDuplicateKeyException (BC-2961)

Issue – If term number "1" already exists for the policy, sending a Rewrite billing instruction without specifying the term number will throw a DBDuplicateKeyException because BillingCenter uses the number "1" for term number.

Workaround – Include a term number that is unique for the given policy number when sending a rewrite billing instruction.

Adding an existing contact following a new contact causes the added contact not to display on New Account and New Producer list views (BC-881)

Issue – Attempting to add an existing contact following a new contact causes the added contact to not display properly in the list view on the Add Contact wizard on the New Account and New Producer screens.

Workaround – Click on a column heading to sort the contacts. Sorting causes the screen to refresh and display the contacts correctly.

Reversing write-off commissions and write-off commission recovery not in the user interface (BC-7184)

Issue – The BillingCenter user interface does not provide a way to reverse write-off commissions or write-off commission recovery.

Workaround – Clone the Reverse button under the Payments tab and reverse `CmsnPayableReduction` and other transactions using the following action: `CurrentLocation.startEditing()`, `producerTransaction.reverse()`, and `CurrentLocation.commit()`.

Studio/Platform Issues

Chrome browser cannot display product documentation in HTML format (DOC-7251)

Issue – If you use the Google Chrome browser, you can view the HTML Guidewire product documentation only if it is served by an HTTP server using the `http://` protocol. The Chrome browser cannot load HTML product documentation from your local disk by using the `file://` protocol.

Workaround – Use a different browser, such as Microsoft Internet Explorer. If you use the HTML documentation only to access the PDF files, you can open them directly in the `pdf` subdirectory of the `doc` directory.

GX model generated XSD cannot be parsed by JAXB (PL-13598)

Issue – XSD generated by the GX model cannot be parsed by JAXB.

Workaround – Add JAXB annotation elements to the XSD to specify the necessary metadata, such as class names, to enable JAXB to generate the Java class files. Contact Guidewire Support for a sample XSD file that is annotated this way.

Problem with regen-java-api command and JAR files (PL-16351)

Issue – If you run the `gwbc regen-java-api` command, BillingCenter creates a `BillingCenter/java-api/lib` directory and puts JAR files for Java development in this directory. However, BillingCenter does not always perform this task correctly, resulting in some of the generated JAR files not being copied to this directory.

Workaround – For missing library files, copy or import into your Java development environment the missing library files from the following directory:

`BillingCenter/admin/lib`

Copy them into the following directory:

`BillingCenter/java-api/lib`

Renaming method or property throws ParseResultsException (PL-16633)

Issue – If you rename a property or a method or you change a method signature, and a workflow references that property or method in a Gosu field, BillingCenter throws `ParseResultsException`. This is the intended behavior.

Workaround – Restart the workflow engine. To do so:

1. Log in to BillingCenter using an administrative account.
2. Access Internal Tools → Reload.
3. Click Reload Workflow Engine.

Gosu class can override @InternalAPI methods when a PublishInGosu java class is subclassed (PL-18217)

Issue – When a Gosu class extends a Java class, it is possible for the Gosu class to override methods in the Java class that are marked as `@InternalAPI`. These overrides could lead to unpredictable behavior.

Workaround – Do not override methods marked as `@InternalAPI` when creating a Gosu class that subclasses a Java class.

Client-side document production scripts cannot be customized in this release (PL-21502)

Issue – In previous releases, client-side document production scripts downloaded by the ActiveX Document Assistant could be customized by modifying JScript files in the web application and removing the cached copies from a `temp` directory on all user computers. In this release, the ActiveX control was replaced by a signed Java Web Start (JWS) application. Because client-side scripts are encapsulated in the signed JWS application, you cannot change the scripts in this release.

Workaround – Guidewire is aware of this issue.

New inbound integration system requires additional configuration information (PL-25227)

Issue – This release includes a new inbound integration system, which is documented in the BillingCenter Integration Guide. Additional configuration information is necessary to use the new API in this release.

Workaround – Contact Guidewire Customer Support for details.

JBoss 6 application server unable to start (PL-27203)

Issue – JBoss 6 generates an exception when it is deployed with BillingCenter 8.0.0.

Workaround – Remove or comment out the tag `<resource-ref>` in the file `web.xml`.

Multiple rule folders are created during a configuration upgrade (PL-27338)

Issue – Multiple rule folders are created if you repeatedly run the configuration upgrade tool followed by the `clean` command.

Workaround – Restore the innermost rules folder to its proper location.

1. Copy the innermost folder to a temporary location.
2. Remove all the nested folders from the original location.
3. Copy back the innermost folder from the temporary location to its proper location.
4. Make sure the folder is named correctly (`BillingCenter8_0_0Rules`).

List view columns that are initially not visible and then set to visible always appear on the right side (PL-27556)

Issue – In BillingCenter 8.0, you can reorder the columns of list views or change their width. These settings are then saved by BillingCenter in the web browser for each list view, so the same ordering and width can be used when the page is revisited.

However, a layout of a list view can change due to differences in data or because the server configuration has changed. New columns added to the list view since the last time the user visited show up on the far right side of the list view, and not in the order specified in the PCF file. This can be confusing, especially when you must scroll the page to the right to see the new columns.

This behavior can also occur when there are two modes of a PCF page containing list views of a similar structure, but with a different ordering of columns. The order and width settings can be applied to the wrong list view in this case, and columns can appear in a different order than intended.

Workaround – If list view columns seem to be missing, first scroll to the right to see if they are there. To correct the order of list view columns, you can reset your layout preferences to restore the default list ordering and widths. To do this, select **Options** → **Clear Layout Preference** in Guidewire BillingCenter.

Javadoc command does not generate index file in expected location (PL-27679)

Issue – When you run `gwbc regen-java-api`, an `index.html` file is not created in `BillingCenter/java-api/doc`.

Workaround – The command now generates Javadoc JAR files in `BillingCenter/java-api/doc`. To view the Javadoc, add the JAR files to the Studio project.

Command to generate data dictionary fails if maxSPVInclusions option is specified (PL-27693)

Issue – The data dictionary is not generated when you run the `gwbc regen-dictionary` command with the `maxSPVInclusions` option.

Workaround – Do not use the `maxSPVInclusions` option with this command.

Gosu does not automatically downcast if the left side of the typeis or typeof expression uses deprecated members (PL-27724)

Issue – To improve readability of your Gosu code, Gosu automatically downcasts after a `typeis` expression if the type is a subtype of the original type. This is particularly valuable for `if` statements and similar Gosu structures. For example, if a variable has type `Object`, you can use code such as:

```
if( x typeis String ) {  
    length // NOTE: length is a property on String, but *not* on Object.  
}
```

In this release, Gosu does not automatically downcast if the left side of the `typeis` or `typeof` expression uses deprecated members. This may result in new compilation errors.

Workaround – To fix these compiler errors, explicitly downcast with the “as” keyword before you access properties or methods on the subtype but not on the original type. For example, suppose a property called `Dep` is deprecated:

```
if (x.Dep typeis ExampleType) {  
    return (x.Dep as ExampleType).PropertyOnExampleSubtype  
}
```

Upgrade trigger for postOnChange on PCF widgets is not working in some cases (PL-27755)

Issue – In PCF files for some widgets, the BillingCenter 8.0.0 upgrade tool does not upgrade the `postOnChange` property to the new syntax for this property. This problem can occur with any widgets that you have added that use `postOnChange` and with any widgets in the base configuration for which you have set `postOnChange`.

Workaround – After running the BillingCenter 8.0.0 upgrade tool:

1. Find all instances of widgets that did not have their `postOnChange` properties converted. For example, search the `BillingCenter/modules/configuration/config/web/pcf` folder and subfolders for occurrences of `postOnChange=`. The instances that you will find are either widgets that need correction or widgets that are commented out (disabled.) There is no need to make the correction on disabled widgets, although there is also no harm in doing so.
2. For each widget that needs correction, open its PCF file in an XML editor and change the following old syntax to the new syntax:
 - Old syntax example:

```
<Input id="xxx"  
      postOnChange="true"  
      onChange="someMethod()  
      disablePostOnEnter="doEvaluation()"/>
```
 - New syntax example:

```
<Input id="xxx">  
  <PostOnChange onChange="someMethod()" disablePostOnEnter="doEvaluation()"/>  
</Input>
```

3. To verify that you have corrected all instances, open Guidewire Studio and navigate in the Project window to configuration → config → Page Configuration. Then compile all files in the pcf folder. If there are no errors relating to postOnChange, your corrections are complete.

New mechanism for reloading Gosu classes (DOC-8218)

Issue – In past releases, you could modify your PCF files and Gosu classes in Studio, and then reload the changes into the running server by pressing Alt+Shift+L in the application user interface. This shortcut no longer loads Gosu classes.

Workaround – To have the server reload your Gosu classes, in Studio, click **Build** → **Make Project**. When Studio is finished compiling your project, the changes will be loaded. You can also restart your server to load the Gosu classes.



chapter 21

Guidewire BillingCenter 7.0.7 Release Notes

Release 7.0.7

Document Revision: 02-July-2015

Copyright © 2001-2015 Guidewire Software, Inc. All rights reserved.

Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire Reinsurance Management, Guidewire ContactManager, Guidewire Vendor Data Management, Guidewire Client Data Management, Guidewire Rating Management, Guidewire InsuranceSuite, Guidewire ContactCenter, Guidewire Studio, Guidewire Product Designer, Guidewire Live, Guidewire DataHub, Guidewire InfoCenter, Guidewire Standard Reporting, Guidewire ExampleCenter, Guidewire Account Manager Portal, Guidewire Claim Portal, Guidewire Policyholder Portal, Gosu, Deliver Insurance Your Way, and the Guidewire logo are trademarks, service marks, or registered trademarks of Guidewire Software, Inc. in the United States and/or other countries.

All other trademarks are the property of their respective owners.

This material is confidential and proprietary to Guidewire and subject to the confidentiality terms in the applicable license agreement and/or separate nondisclosure agreement.

Guidewire products are protected by one or more United States patents.

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Support
- Major Issues and Changes
- Improvements and Resolved Issues
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire BillingCenter.

IMPORTANT If you skipped one or more upgrade releases to BillingCenter, be sure to read the Release Notes for those releases to learn about changes and fixed issues.

Version Number

This release of Guidewire BillingCenter is 7.0.7.

- If you are integrating with Guidewire PolicyCenter, this version of BillingCenter requires PolicyCenter 7.0.4 or later.
- If you are integrating with Guidewire ContactManager, this version of BillingCenter requires ContactManager 7.0.0 or later. The latest maintenance release is preferred.

Installing This Release

IMPORTANT If you use an Oracle database, this release requires Oracle patch 12813641.

Refer to the following:

- The *BillingCenter Installation Guide* for general installation information.
- Prior BillingCenter Release Notes for any versions that you have skipped.

Support

For assistance, visit the Guidewire Resource Portal – <http://guidewire.custhelp.com>

Major Issues and Changes

This section contains issues and major changes that may affect your installation.

- New Version of Guidewire Document Assistant ActiveX Control (PL-29937)
- Upgrade Support
- Base PCF File Changes
- Changes in This Release Provided in Upgrade Diff Report

New Version of Guidewire Document Assistant ActiveX Control (PL-29937)

This release includes a new version of the Guidewire Document Assistant ActiveX control. This new version fixes an issue that required users to have administrative privileges for installation.

Upgrade Support

- Upgrading from BillingCenter 2.1.9 or newer versions is supported in this release.

- BillingCenter 7.0.4 and later include multiple enhancements to improve performance. As a result of some of these improvements, the upgrade process of a large BillingCenter database (one million or more policies per year) requires more processing time than usual for a maintenance release. For more information, see issue **BC-12707** in the section “BillingCenter Known Issues.”
- The following information is not yet included in the BillingCenter Upgrade Guide:
In cases where the definition of an Entity has been changed from being retireable to being non-retireable, BillingCenter 7.0.3 and later versions will not automatically remove data rows that had been retired. Instead, before upgrading to BillingCenter 7.0.3 or later versions, a new version check, `MigrateTableFromRetireableToEditableVersionCheck`, will run. This version check will log an error if any retired rows are found. If any such errors are logged, the retired rows must be examined and removed or restored before the upgrade can continue.

Base PCF File Changes

BillingCenter release 7.0.6 to 7.0.7

- There are no changes to the base PCF files.

Changes in This Release Provided in Upgrade Diff Report

Guidewire provides a report detailing certain differences between the current release and your prior release. This report describes changes in display keys, entities, typelists, and the Gosu API. To obtain your custom Upgrade Diff Report, visit the Guidewire Resource Portal at <https://guidewire.custhelp.com>.

Improvements and Resolved Issues

BillingCenter Improvements and Resolved Issues

ID	Description
Database Upgrader	
BC-9058	<p>Upgrading of the following data type is supported:</p> <ul style="list-style-type: none">• Policy period data created by performing multiple transfers between a group of Producer Codes. For example, invoking the <code>Po1TransferByProdCode</code> method multiple times to perform the following sequence of transfers:<ul style="list-style-type: none">• Producer A → Producer B• Producer B → Producer C• Producer C → Producer B, at which point Producer B has received a transfer from both Producer A and Producer C. <p>In previous releases, data created in this manner could not be upgraded.</p>
BC-14106	Fixed a condition in which a bulk update on <code>bc_producer</code> generated unnecessary updates. The updates occurred when pending values canceled each other, resulting in a zero value.
BC-14163	Improved the performance of the Upgrade Commission batch process. This improvement is relevant when moving from BillingCenter 2.1 to 7.0.x.
BC-14188	Improved the performance of <code>DirectBillDistVersionTrigger</code> . This improvement is relevant when upgrading from BillingCenter 2.1 to 7.0.x.
BC-14189	Improved the performance of <code>InvoiceItemCommissionVersionTrigger</code> . This improvement is relevant when upgrading from BillingCenter 2.1 to 7.0.x.
BC-14207	Improved the performance of the <code>ItemCommission</code> upgrade trigger. This improvement is relevant when upgrading from BillingCenter 2.1 to 7.0.x.
BC-14208	Improved the performance of the set of upgrade triggers that transform <code>ChargePaid</code> transactions. This improvement is relevant when upgrading from BillingCenter 2.1 to 7.0.x.

ID	Description
BC-14211	Improved the performance of <code>InvoiceItemHasBilledAndDueTxnTrigger</code> . This improvement is relevant when upgrading from BillingCenter 2.1 to 7.0.x.
BC-14242	Upgrades succeed if <code>MigrateToLargeIDsAndDatetime2</code> is TRUE. This solves an issue in BillingCenter 7.0.6.
BC-14291	Improved the performance of <code>AddStatementInvoiceAllInvoiceItemsExactlyPaidVersionTrigger</code> . This improvement is relevant when upgrading from BillingCenter 2.1 and later to 7.0.x.
BC-14296	When converting SQL Server primary keys from <code>integer</code> to <code>bigint</code> or from <code>datetime</code> to <code>datetime2</code> , the conversion process completes in less time than earlier releases.
BC-14344	Improved the performance of <code>NegativeChargeBilledTrigger</code> . This improvement is relevant when upgrading from BillingCenter 2.1 and later to 7.0.x.
General Application Issues	
BC-14022	Fixed a condition in which a check against an incorrect date parameter resulted in database consistency failures.
BC-14338	Fixed an issue where incremental updates could be duplicated to denorm values if the bundle was rolled back late in the commit cycle.
Suspense Payment	
BC-14122	Fixed an issue that enabled reversed suspense payments to be applied to an account, policy, or producer.
BC-14124	Fixed an issue where reversing a suspense payment that had been edited did not reverse all relevant transactions.
Transaction Transfer	
BC-14150	A policy can be transferred to another account without also moving the invoice items associated with the policy. In earlier releases, transferring a policy to another account also moved the policy's invoice items to the account. Note: This ability is not intended for general use. It is provided for special situations only and may not be available in future releases.

Known Issues and Limitations

This section describes known issues with this release of Guidewire BillingCenter.

- BillingCenter Known Issues
- Studio/Platform Known Issues

Note: For maintenance releases, Guidewire often defers fixing configuration issues that require merging files during the upgrade. Workarounds to many of these issues are listed in the following topics. The goal of this policy is to make upgrades as straightforward as possible.

BillingCenter Known Issues

Performance improvements cause upgrade to take longer than usual in a maintenance release [\(BC-12707\)](#)

Issue – BillingCenter 7.0.4 and later include enhancements that will result in better performance, especially for large databases. As a result of these enhancements, large-volume customers will notice that the upgrade process takes longer than most maintenance releases.

Guidewire internal tests using a database containing one million policies with one year of data took one hour to upgrade on SQL Server. There should be minimal impact to customers with less than one million policies.

Workaround – Guidewire recommends performing a test upgrade in an environment that simulates your production environment, including using similar hardware and a similarly sized database.

For assistance in estimating how long an upgrade will take, please contact Guidewire Customer Support.

Delinquency: Delinquency can be started on an account, even though there is no amount due, when negative charges are present (BC-3746)

Issue – Delinquency can be started on an account when no amount is due if a negative charge on a policy is present.

Workaround – Invoke the `doDueTransaction()` method on the negative invoice item. This will execute a `ChargeDue` transaction for that invoice item which will prevent the problem.

Note that you can execute `invoiceItem.doDueTransaction()` method on a negative invoice item even if it is on an invoice that is in billed state not yet due. Later, when the invoice is made due, BillingCenter will just skip the negative invoice (because `invoiceItem.doDueTransaction()` already forced that invoice item to be due).

Contact Manager: BC does not show updated Primary Address after CM contact Merge (BC-9315)

Issue – If you are using BillingCenter with Contact Manager, there is a problem with merging contacts with different addresses. The primary address of the merged contact is not shown in BillingCenter.

Workaround – Guidewire is aware of this issue.

Commission: Commission override after a commission write-off causes write-off to be reserved (BC-9536)

Issue – Several methods of the `BillingAPI` WS-I webservice, such as `issuePolicyPeriod()` and `cancelPolicyperiod()`, indirectly call `BillingInstructionUtil.executeAndCommit()` which commits the current bundle.

These methods also directly call `tryCatch()`, which commits a new bundle after executing the entire block of code inside of `tryCatch()`. As a result, the code tries to commit the same bundle twice.

Here are two ways you can encounter this problem:

1. If you try to create an activity within the block of code passed to `tryCatch()` after the call to execute the billing instruction, BillingCenter will throw a `DBAAlreadyExecutedException`.

For example:

```
function issuePolicyPeriod(issuePolicyInfo : IssuePolicyInfo, transactionId : String) : String{
    require(issuePolicyInfo, "issuePolicyInfo")
    var publicID = tryCatch( \ bundle -> {
        var pid= issuePolicyInfo.executeIssuanceBIC()
        // code *here* to create a new activity
        return pid
    }, transactionId)
    return publicID
}
```

2. If you used `CreateTroubleTicketHelper` to create a trouble ticket in a billing instruction preupdate rule, you would again get a `DBAAlreadyExecutedException`. The following is an example of a billing instruction preupdate rule that would trigger the error:

```
// create trouble ticket with a delq hold until second invoice
var troubleTicketHelper = new CreateTroubleTicketHelper()
var ticket = troubleTicketHelper.createTroubleTicket((billingInstruction as
    PlcyBillingInstruction).PolicyPeriod)
ticket.Title = "Delinquency hold on issuance"
ticket.DetailedDescription = ticket.Title
ticket.Priority = "normal"
ticket.TicketType = "automatic"
ticket.Hold.setAppliedToHoldType(HoldType.TC_DELINQUENCY, true)
ticket.Hold.setReleaseDate(HoldType.TC_DELINQUENCY, gw.util.GWBaseDateEnhancement.CurrentDate)
```

Workaround –

1. For this case, in the block of code passed to `tryCatch()`, do not create or update entities after the call to execute the billing instruction. For example:

```
function issuePolicyPeriod(issuePolicyInfo : IssuePolicyInfo, transactionId : String) : String{
    require(issuePolicyInfo, "issuePolicyInfo")
```

```

var publicID = tryCatch( \ bundle -> {
    // code *here* to create a new activity
    var pid= issuePolicyInfo.executeIssuanceBI()
    return pid
}, transactionId)
return publicID
}

```

2. For this case, instead of using `CreateTroubleTicketHelper` to create the trouble ticket, create related entities directly as in the following example in a PolicyPeriod enhancement:

```

var ticket = new TroubleTicket(this)
var ttJoinEntity = new TroubleTicketJoinEntity(ticket)
ticket.Title = ticketTitle
ticket.DetailedDescription = "Delinquency hold placed on Account with a " + ticket.Title
ticket.Priority = "normal"
ticket.TicketType = ticketType
ticket.Hold.setAppliedToHoldType( HoldType.TC_DELINQUENCY, true )
ticket.Hold.setReleaseDate(HoldType.TC_DELINQUENCY, releaseDate)
ticket.addToTroubleTicketJoinEntities(ttJoinEntity)
this.addToTroubleTicketJoinEntities(ttJoinEntity)
this.Account.addToTroubleTicketJoinEntities(ttJoinEntity)
ticket.Hold.checkForHoldAdditions()

```

[PolicyCenter Integration: BC Renewal thread fails with exception on issue after removal of Alt Billing Account from Renewal in PC \(BC-9540\)](#)

Issue – If you are integrated with PolicyCenter, you will encounter an error if you renew a policy that had an alternate billing account and you remove the alternate billing account in PolicyCenter prior to committing renewal.

Workaround – Guidewire is aware of this issue.

[PolicyCenter Integration: On Renewal Offer, Suspense Payment Should Be Applied At Policy Period \(BC-9691\)](#)

Issue – When integrated with PolicyCenter, a suspense payment that is created with an offer number from a PolicyCenter renewal is getting applied to the owner account of the policy period. The suspense payment should be applied to the renewal policy period instead of the owner account.

Workaround – Guidewire is aware of this issue.

[PolicyCenter Integration: Policy Issuance Generates Benign Log Errors \(BC-14032\)](#)

Issue – Issuing a policy period in PolicyCenter 8.0.2 generates `BadIdentifierException` errors in the logs of BillingCenter 7.0.6 and later and ContactManager 8.0.2. They are generated because the contact does not exist in BillingCenter and, thus, could not be found there.

Workaround – The log errors should not be generated and can be safely ignored. The generation of the errors is repaired in BillingCenter 8.0.2.

[Payments: Reversing Producer Payments is throwing NullPointerException when the payment is related to a retired invoice item \(BC-9749\)](#)

Issue – BillingCenter does not allow invoice items that have earned commission to be deleted. BillingCenter already prevents such items from being removed through the user interface. If an invoice item that has earned commission is deleted programmatically, errors will occur if the related payment item is reversed.

Workaround – Instead of deleting invoice items that have earned commission, set the invoice item amount to zero.

Disbursement: Changing the Disbursement reason from Automatic causes it to be sent out despite a hold (BC-9761)

Issue – The Disbursements batch process sends an disbursement if the disbursement reason is changed from 'Automatic' to something else, even if there is an active Trouble Ticket with a hold of type 'Disbursements' on the owning account. The BillingCenter user interface does not allow editing of automatic disbursements, but they can be edited through configuration. BillingCenter fails to account for this and the disbursement batch process treats a reason other than automatic to mean that it is a manual disbursement and so processes it, ignoring any holds.

Workaround – Use a custom extension to store more specific reasons for an automatic disbursement.

Payment Distribution: BC should not distribute payment amount that have been disbursed (BC-9783)

Issue – Reversing a DirectBillPayment is causing T-account Unapplied to become negative. This can happen when another direct bill payment on the same account gets modified.

Workaround – Create a recapture charge when unapplied goes below zero.

Changing Billing Method From AB to DB for earning on First Payment Received For Billed Statement Is Earning Commission (BC-10476)

Issue – Changing the billing method on a policy period from agency bill to direct bill with a producer that earns commission on First Payment Received will instead earn the commission when invoiced. Commission should not be earned until a payment is received.

Workaround – As part of transferring a policy period from agency bill to direct bill, add code to the policy period preupdate rule to reverse the earning transaction if the total amount paid on the policy period is zero. This will make the earning go back to zero for the policy period. The commission payable batch process will not identify these charge commissions (since their earning criteria will not match the selection criteria). When a real payment is posted to the policy, commission earning will be triggered by that event and be earned correctly.

PolicyNumberLongDenorm not updated in upgrade (BC-10742)

Issue – Prior to 3.0, BillingCenter used 0-based term numbers for policies. Starting with 3.0, BillingCenter uses 1-based term numbers for policies to be consistent with PolicyCenter term numbering. This issue affects upgrades from BillingCenter versions prior to 3.0 only.

Because the `PolicyNumberLong` field (and the corresponding `PolicyNumberLongDenorm` field) can be a legal reference, BillingCenter does not change the value during upgrade.

The upgrade does renumber the `TermNumber` from being 0-based to being 1-based, since the `TermNumber` is not normally used as a legal reference. This behavior is to avoid a gap between term numbers. Otherwise, the terms would have a gap. For example, instead of `TermNumber` 0, 1, <upgrade> 3, 4, the upgrade renames the first two to make it 1, 2, 3, 4.

However, because the upgrade does not adjust `PolicyNumberLong` and `PolicyNumberLongDenorm`, there will be a gap in the `PolicyNumberLong` (for example 123456-0, 123456-1, <upgrade> 123456-3, 123456-4, skipping any 123456-2 `PolicyNumberLong`).

Workaround – If you do not need to preserve your policy numbers after upgrade and you want your `PolicyNumberLong` and `PolicyNumberLongDenorm` values to match the new post-upgrade `TermNumber`, contact Guidewire Customer Support for information on how to update these fields.

IMPORTANT: The `PolicyNumberLong` and `PolicyNumberLongDenorm` columns must be kept in sync.

UpgradeCommission batch process is throwing an error when a reversal or payment plan refactor happens after a point in time transfer (BC-10917)

Issue – Under specific circumstances where a point in time transfer occurs, then a reversal or payment plan refactor occurs, then the BillingCenter database is upgraded to version 7.0.3, an error can later occur when running the UpgradeCommission batch process.

Workaround – Contact Guidewire Customer Support for assistance if you encounter this issue.

Reversing Write-Off Commissions and Write-Off Commission Recovery in the UI (BC-7184)

Issue – The BillingCenter user interface does not provide a way to reverse Write-Off Commissions and Write-Off Commission Recovery.

Workaround – Clone the Reverse button under the Payments tab and reverse CmsnPayableReduction and other transactions using the following action: CurrentLocation.startEditing(); producerTransaction.reverse(); and CurrentLocation.commit().

Rewrite BI with the same policynumber as the original policy is not incrementing the TermNumber and throws DBDuplicateKeyException (BC-2961)

Issue – Sending a rewrite billing instruction without specifying the term number will throw a DBDuplicateKeyException if term number “1” already exists for the policy. If you omit the term number, BillingCenter uses the number “1” for term number. If a policy period with the term number “1” already exists then BillingCenter will throw an exception.

Workaround – Include a term number that is unique for the given policy number when sending a rewrite billing instruction.

AGBL statement Total Payment Applied is not correct after change 1 of 2 producer policies payment schedule (BC-2966)

Issue – The producer statement does not show the correct dollar amount after a change is made to one of the policy’s payment plans on a policy with two producers after the statement was billed and partially paid.

Workaround – Guidewire is aware of this issue.

Upgrade from BillingCenter 2.1.10 with an agency bill statement with a commission remainder invoice item may result in incorrect written off and net amounts for that agency bill cycle (BC-10756)

Issue – If you are upgrading from BillingCenter 2.1.10 and you have an agency bill statement with a commission remainder invoice item, the upgraded written off and net amounts on the agency bill cycle can be inaccurate. The agency bill statement will correctly reflect the commission remainder. Only the agency bill cycle has the incorrect (not adjusted by commission remainder write-off) amounts for net and written off amounts.

Workaround – Guidewire is aware of this issue.

Studio/Platform Known Issues

Problem with regen-java-api command and JAR files (PL-16351)

Issue – If you run the `BillingCenter/bin/gwpc regen-java-api` command, BillingCenter creates a `BillingCenter/java-api/lib` directory and puts JAR files for Java development in this directory. However, BillingCenter does not always perform this task correctly, resulting in some of the generated JAR files not being copied to this directory.

Workaround – For missing library files, copy or import into your Java development environment the missing library files from the following directory:

BillingCenter/admin/lib

Copy them into the following directory:

BillingCenter/java-api/lib

[Renaming method or property throws ParseResultsException \(PL-16633\)](#)

Issue – If you rename a property or a method or you change a method signature, and a workflow references that property or method in a Gosu field, BillingCenter throws `ParseResultsException`. This is the intended behavior.

Workaround – Restart the workflow engine. To do so:

1. Log into BillingCenter using an administrative account.
2. Access **Internal Tools** → **Reload**.
3. Click **Reload Workflow Engine**.

[In some languages, web browsers render column headers of list views improperly \(PL-18027\)](#)

Issue – In some languages, web browsers render some column headers of list views improperly if their column widths are specified too narrowly in their PCF definitions. For example, sometimes a numeric column is specified with a variable width of 1%. This narrow setting forces the browser to render the column too narrowly for the text of the translated column heading.

Workaround – Edit the PCF file that defines the column and clear the value from the width property. Without a specified value for the column width, browsers render the column widely enough to display the full text of the translated column heading.



chapter 22

Guidewire BillingCenter 7.0.6 Release Notes

Release 7.0.6

Document Revision: 02-July-2015

Copyright © 2001-2015 Guidewire Software, Inc. All rights reserved.

Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire Reinsurance Management, Guidewire ContactManager, Guidewire Vendor Data Management, Guidewire Client Data Management, Guidewire Rating Management, Guidewire InsuranceSuite, Guidewire ContactCenter, Guidewire Studio, Guidewire Product Designer, Guidewire Live, Guidewire DataHub, Guidewire InfoCenter, Guidewire Standard Reporting, Guidewire ExampleCenter, Guidewire Account Manager Portal, Guidewire Claim Portal, Guidewire Policyholder Portal, Gosu, Deliver Insurance Your Way, and the Guidewire logo are trademarks, service marks, or registered trademarks of Guidewire Software, Inc. in the United States and/or other countries.

All other trademarks are the property of their respective owners.

This material is confidential and proprietary to Guidewire and subject to the confidentiality terms in the applicable license agreement and/or separate nondisclosure agreement.

Guidewire products are protected by one or more United States patents.

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Support
- Issues and Major Changes
- Improvements and General Issues
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire BillingCenter.

Version Number

This release of Guidewire BillingCenter is 7.0.6.

- If you are integrating with Guidewire PolicyCenter, this version of BillingCenter requires PolicyCenter 7.0.4 or later.
- If you are integrating with Guidewire ContactManager, this version of BillingCenter requires ContactManager 7.0.0 or later. The latest maintenance release is preferred.

Installing This Release

IMPORTANT If you use an Oracle database, this release requires Oracle patch 12813641.

Refer to the following:

- The *BillingCenter Installation Guide* for general installation information.
- Prior BillingCenter release notes for any versions that you have skipped.

Support

For assistance, visit the Guidewire Resource Portal – <http://guidewire.custhelp.com>

Issues and Major Changes

This section contains issues and major changes that may affect your installation.

- Upgrade Support
- Base PCF File Changes
- Changes in This Release Provided in Upgrade Diff Report

Upgrade Support

- Upgrading from BillingCenter 2.1.9 or newer versions is supported in this release.
- This release includes multiple enhancements that will improve performance. As a result of some of these performance improvements, upgrade of large BillingCenter databases (one million or more policies per year) will take longer than upgrading to earlier maintenance releases. For more information see **BC-12707** under “BillingCenter Known Issues.”
- The following information is not yet included in the BillingCenter Upgrade Guide:
In cases where the definition of an Entity has been changed from being retireable to being non-retirable, BillingCenter 7.0.3 and later versions will not automatically remove data rows that had been retired. Instead, before upgrading to BillingCenter 7.0.3 or later versions, a new version check, `MigrateTableFromRetireableToEditableVersionCheck`, will run. This version check will log an error if any retired rows are found. If any such errors are logged, the retired rows must be examined and removed or restored before upgrade can continue.

Base PCF File Changes

All links below require the `readme_files` directory on your local disk.

BillingCenter release 7.0.5 to 7.0.6

- To view a report of the changes in the base PCF files in the `modules/bc` directory, [click here](#).
- To view a report of the changes in the base PCF files in the `modules/p1` directory, [click here](#).

Changes in This Release Provided in Upgrade Diff Report

Guidewire provides a report detailing certain differences between the current release and your prior release. This report describes changes in display keys, entities, typelists, and the Gosu API. To obtain your custom Upgrade Diff Report, visit the Guidewire Resource Portal.

Improvements and General Issues

BillingCenter Improvements and General Issues

The following are the improvements added to and the general issues corrected in this release:

ID	Description
Accounting Subsystem	
BC-13224	<p>Added the configuration parameter <code>EnableChargeProRataTxCreation</code> to disable creation of <code>ChargeProRataTx</code> entities. BillingCenter normally creates <code>ChargeProRataTx</code> entities when a charge is created with a <code>ChargePattern</code> subtype of <code>ProRataCharge</code>. The entities are subsequently processed by the <code>ChargeProRataTx</code> batch process to track earned premium. If you do not use BillingCenter to track earned premium, the <code>ChargeProRataTx</code> entities are not needed. When <code>EnableChargeProRataTxCreation</code> is set to <code>false</code>, <code>ChargeProRataTx</code> entities are not created.</p> <p>In BillingCenter 7.0.6, the default value of <code>EnableChargeProRataTxCreation</code> is <code>true</code>. If <code>EnableChargeProRataTxCreation</code> is set to <code>false</code>, it cannot be enabled again at a later time.</p> <p>Note: The <code>EnableChargeProRataTxCreation</code> parameter and <code>ChargeProRataTx</code> batch process are used only to track earned premium. If you do not use BillingCenter to track earned premium, set <code>EnableChargeProRataTxCreation</code> to <code>false</code>, and do not run the <code>ChargeProRataTx</code> batch process. Doing so reduces data storage usage and improves system performance. Any existing <code>ChargeProRataTx</code> entities not processed by the <code>ChargeProRataTx</code> batch process are ignored by BillingCenter.</p>
Agency Bill	
BC-13322	<p>On the <code>Producer->Agency Bill Payments</code> screen, the base configuration does not show distributed items that have been reversed. To show reversed items, call the new method <code>AgencyCycleDistView.withRetrieveOnlyReversedDistItems</code>. Note: This new method is applicable only to BillingCenter 7.0.5p1 and 7.0.6. In BillingCenter 8, reversed items can be shown by changing the screen's PCF file.</p>
Commission / Commission Plan	
BC-12984	Fixed the allocation of commission writeoff items when some items have mixed charges of reserve and payable. Mixed charges include positive and negative charges or over and under settled items.
Database Upgrader	
BC-11654	<p>Before this release, upgrading from BillingCenter 3 or BillingCenter 7 did not automatically add new permissions to the Superuser and General Admin roles.</p> <p>This issue has been fixed. It is no longer necessary to manually update the Superuser and General Admin roles after your database upgrade is complete.</p>
Disbursement	
BC-13239	<p>Significant performance improvement achieved when running the <code>New Payment</code> and <code>Automatic Disbursement</code> batch processes.</p> <p>If you have a large database, the upgrade may take longer than normal for a maintenance release. This is because new indexes and histograms are being created for several tables, specifically <code>bc_account</code>, <code>bc_taccount</code>, <code>basemoneyreceived</code>, and <code>paymentmoneyreceived</code>. Creating the new histogram on <code>bc_taccount.BalanceDenorm</code> can be particularly expensive on Oracle databases. For best results, set your database statistics configuration so <code>samplingpercentage</code> is 0 and <code>databasedegree</code> is greater than 1.</p>
Delinquency / Delinquency Plan	
BC-11736	<p>On the <code>Desktop->My Delinquencies</code> page, the default filter is changed from <code>All open</code> to <code>New opened (this week)</code>. The former default filter could potentially return a large number of results, negatively affecting performance.</p> <p>Also on the <code>My Delinquencies</code> page, the filters <code>All open owned</code> and <code>Closed in last 30 days</code> have been removed and deprecated. In the base configuration, these filters never returned any results. If these filters have been configured in your system to return results, contact Guidewire Customer Support for assistance.</p>

ID	Description
BC-13151	Account-level delinquencies can be loaded into BillingCenter by using staging tables. In earlier versions, only <code>DeInquencyProcess</code> rows of subtype <code>PolicyDlnqProcess</code> could be loaded. Now the account-level subtype <code>DeInquencyProcess</code> can also be loaded. In addition, the <code>LegacyDeInquency</code> work queue now traverses both of these subtypes after the integrity-check-and-load phase.
General	
BC-11077	When a disbursement is rejected, the status of both the disbursement and the approval are updated, accordingly. In earlier versions, the approval status was not updated.
BC-12256	Modifications to an executed payment or credit distribution may not be performed directly. For example, an executed distribution cannot be modified by calling the <code>BaseDistImpl</code> methods to alter individual components of the distribution. Attempts to directly change executed payment or credit distributions throw an exception. To change an executed payment distribution, make a copy of the distribution and then modify and execute the copy.
Invoice/Charge	
BC-10505	When an invoice is carried forward, its items are placed on a planned invoice within the same invoice stream. In earlier versions, the items might have been placed on a planned invoice within any of the account's invoice streams.
Platform	
BC-13019	If you are using an Oracle database, you must apply Oracle patch 12813641 to resolve an Oracle issue. Failure to apply this patch could result in errors if the database statistics are updated while BillingCenter work queues are running or data is being updated through the BillingCenter user interface.
Suspense Payment	
BC-13670	To successfully reverse a suspense payment item, a corresponding unapplied suspense payment transaction or suspense release transaction must exist. If such a transaction does not exist, the reverse operation aborts and throws an <code>IllegalStateException</code> . In previous releases, BillingCenter incorrectly allowed the reversal of any suspense payment item.
BC-14008	Situations existed in previous releases that allowed a suspense payment to be applied multiple times. A suspense payment can now be applied only if a non-reversed suspense payment transaction exists.
BC-14057	While editing a payment, an associated suspense item can be modified and released. In earlier versions, modifying and releasing a suspense item while editing a payment would create incorrect data.

Platform Improvements and General Issues

ID	Description
Activities	
PL-28473	A new permission has been added to allow users to assign activities from a queue. This permission is not assigned to any roles by default. To allow users to assign activities from a queue, add the new permission <code>actqueueassign</code> or <code>SystemPermissionType.TC_ACTQUEUEASSIGN</code> to the necessary role.
Clustering	
PL-29170	Added a JGroups watchdog timer that resets cluster nodes if they stop responding to the cluster. This includes the following new configuration parameters: <ul style="list-style-type: none"> <code>JGroupsWatchdogHeartbeatIntervalSecs</code> - The number of seconds between each heartbeat ping from the cluster coordinator to each of the nodes. Default: 30 seconds <code>JGroupsWatchdogMissedHeartbeatsBeforeReset</code> - The number of missed heartbeats after which a node resets its JGroups channel. Default: 10
PL-29237	Changed the default value for the <code>ClusterProtocolStack</code> configuration parameter. For details, see the <i>Configuration Guide</i> .
Database Instrumentation	
PL-30150	The following options, which are selected by default in the base configuration, have been added to the Oracle AWR info page. You can also deselect any of these options when generating an Oracle AWR report: <ul style="list-style-type: none"> Analysis of messaging Analysis of batch processing and work item Analysis of history
Document Management	
PL-30381	Resolves an issue where temporary files created during document upload were not closed and deleted until the server shutdown. If the server ran for a long time, the available file handles in the application server pool became exhausted. With this release, files are closed and deleted after the upload operation completes and file handles are released.
Gosu	
PL-29166	Fixed an error that would occur during a type system refresh.
Messaging	
PL-29999	Eliminated unnecessary locking and unlocking in the <code>MessageRequest</code> plugin when it is not configured.
Queries	
PL-17270	Fixed an issue in which <code>gw.api.database.QueryProcessor.Count</code> was producing an incorrect SQL statement and count result.
Revisioning	
PL-28241	Added a comment with the source of a lock to the <code>SELECT FOR UPDATE</code> statement. This gives a useful context when debugging locking issues, including being able to separate out the calls from messaging rules from the calls from within messaging.
Web - Other	
PL-29418	Fixed a problem that prevented the <code>Admin</code> page from being shown after applying WebSphere patch <code>7.0.0.0-WS-WASJavaSDK-LinuxX64-IFPM98578.pak</code> .

Known Issues and Limitations

This section describes known issues with this release of Guidewire BillingCenter.

- BillingCenter Known Issues
- Studio/Platform Issues

BillingCenter Known Issues

Performance improvements cause upgrade to take longer than usual in a maintenance release (BC-12707)

Issue – BillingCenter 7.0.4, 7.0.5, and 7.0.6 include enhancements that will result in better performance, especially for large databases. As a result of these changes, large-volume customers will notice that upgrade takes longer than most maintenance releases.

Guidewire internal tests using a database containing one million policies with one year of data took one hour to upgrade on SQL Server. There should be minimal impact to customers with less than one million policies.

Workaround – Guidewire recommends performing a test upgrade in an environment that simulates your production environment, including using similar hardware and a similarly sized database.

For assistance in estimating how long upgrade will take, please contact Guidewire Customer Support.

Delinquency: Delinquency can be started on an account, even though there is no amount due, when negative charges are present (BC-3746)

Issue – Delinquency can be started on an account when no amount is due if a negative charge on a policy is present.

Workaround – Invoke the `doDueTransaction()` method on the negative invoice item. This will execute a `ChargeDue` transaction for that invoice item which will prevent the problem.

Note that you can execute `invoiceItem.doDueTransaction()` method on a negative invoice item even if it is on an invoice that is in billed state not yet due. Later, when the invoice is made due, BillingCenter will just skip the negative invoice (because `invoiceItem.doDueTransaction()` already forced that invoice item to be due).

Contact Manager: BC does not show updated Primary Address after CM contact Merge (BC-9315)

Issue – If you are using BillingCenter with Contact Manager, there is a problem with merging contacts with different addresses. The primary address of the merged contact is not shown in BillingCenter.

Workaround – Guidewire is aware of this issue.

Commission: Commission override after a commission write-off causes write-off to be reserved (BC-9536)

Issue – Several methods of the BillingAPI WS-I webservice, such as `issuePolicyPeriod()` and `cancelPolicyperiod()`, indirectly call `BillingInstructionUtil.executeAndCommit()` which commits the current bundle.

These methods also directly call `tryCatch()`, which commits a new bundle after executing the entire block of code inside of `tryCatch()`. As a result, the code tries to commit the same bundle twice.

Here are two ways you can encounter this problem:

1. If you try to create an activity within the block of code passed to `tryCatch()` after the call to execute the billing instruction, BillingCenter will throw a `DBAAlreadyExecutedException`.

For example:

```
function issuePolicyPeriod(issuePolicyInfo : IssuePolicyInfo, transactionId : String) : String{
    require(issuePolicyInfo, "issuePolicyInfo")
    var publicID = tryCatch( \ bundle -> {
        var pid= issuePolicyInfo.executeIssuanceBI()
        // code *here* to create a new activity
        return pid
    }, transactionId)
    return publicID
}
```

- 2.** If you used `CreateTroubleTicketHelper` to create a trouble ticket in a billing instruction preupdate rule, you would again get a `DBAIStreadyExecutedException`. The following is an example of a billing instruction preupdate rule that would trigger the error:

```
// create trouble ticket with a delq hold until second invoice
var troubleTicketHelper = new CreateTroubleTicketHelper()
var ticket = troubleTicketHelper.createTroubleTicket((billingInstruction as
    PlcyBillingInstruction).PolicyPeriod)
ticket.Title = "Delinquency hold on issuance"
ticket.DetailedDescription = ticket.Title
ticket.Priority = "normal"
ticket.TicketType = "automatic"
ticket.Hold.setAppliedToHoldType(HoldType.TC_DELINQUENCY, true)
ticket.Hold.setReleaseDate(HoldType.TC_DELINQUENCY, gw.util.GWBaseDateEnhancement.CurrentDate)
```

Workaround –

- 1.** For this case, in the block of code passed to `tryCatch()`, do not create or update entities after the call to execute the billing instruction. For example:

```
function issuePolicyPeriod(issuePolicyInfo : IssuePolicyInfo, transactionId : String) : String{
    require(issuePolicyInfo, "issuePolicyInfo")
    var publicID = tryCatch( \ bundle -> {
        // code *here* to create a new activity
        var pid= issuePolicyInfo.executeIssuanceBI()
        return pid
    }, transactionId)
    return publicID
}
```

- 2.** For this case, instead of using `CreateTroubleTicketHelper` to create the trouble ticket, create related entities directly as in the following example in a PolicyPeriod enhancement:

```
var ticket = new TroubleTicket(this)
var ttJoinEntity = new TroubleTicketJoinEntity(ticket)
ticket.Title = ticketTitle
ticket.DetailedDescription = "Delinquency hold placed on Account with a " + ticket.Title
ticket.Priority = "normal"
ticket.TicketType = ticketType
ticket.Hold.setAppliedToHoldType( HoldType.TC_DELINQUENCY, true )
ticket.Hold.setReleaseDate(HoldType.TC_DELINQUENCY, releaseDate)
ticket.addToTroubleTicketJoinEntities(ttJoinEntity)
this.addToTroubleTicketJoinEntities(ttJoinEntity)
this.Account.addToTroubleTicketJoinEntities(ttJoinEntity)
ticket.Hold.checkForHoldAdditions()
```

[PolicyCenter Integration: BC Renewal thread fails with exception on issue after removal of Alt Billing Account from Renewal in PC \(BC-9540\)](#)

Issue – If you are integrated with PolicyCenter, you will encounter an error if you renew a policy that had an alternate billing account and you remove the alternate billing account in PolicyCenter prior to committing renewal.

Workaround – Guidewire is aware of this issue.

[PolicyCenter Integration: On Renewal Offer, Suspense Payment Should Be Applied At Policy Period \(BC-9691\)](#)

Issue – When integrated with PolicyCenter, a suspense payment that is created with an offer number from a PolicyCenter renewal is getting applied to the owner account of the policy period. The suspense payment should be applied to the renewal policy period instead of the owner account.

Workaround – Guidewire is aware of this issue.

[PolicyCenter Integration: Policy Issuance Generates Benign Log Errors \(BC-14032\)](#)

Issue – Issuing a policy period in PolicyCenter 8.0.2 generates `BadIdentifierException` errors in the logs of BillingCenter 7.0.6 and later and ContactManager 8.0.2. They are generated because the contact does not exist in BillingCenter and, thus, could not be found there.

Workaround – The log errors should not be generated and can be safely ignored. The generation of the errors is repaired in BillingCenter 8.0.2.

[Payments: Reversing Producer Payments is throwing NullPointerException when the payment is related to a retired invoice item \(BC-9749\)](#)

Issue – BillingCenter does not allow invoice items that have earned commission to be deleted. BillingCenter already prevents such items from being removed through the user interface. If an invoice item that has earned commission is deleted programmatically, errors will occur if the related payment item is reversed.

Workaround – Instead of deleting invoice items that have earned commission, set the invoice item amount to zero.

[Disbursement: Changing the Disbursement reason from Automatic causes it to be sent out despite a hold \(BC-9761\)](#)

Issue – The Disbursements batch process sends an disbursement if the disbursement reason is changed from 'Automatic' to something else, even if there is an active Trouble Ticket with a hold of type 'Disbursements' on the owning account. The BillingCenter user interface does not allow editing of automatic disbursements, but they can be edited through configuration. BillingCenter fails to account for this and the disbursement batch process treats a reason other than automatic to mean that it is a manual disbursement and so processes it, ignoring any holds.

Workaround – Use a custom extension to store more specific reasons for an automatic disbursement.

[Payment Distribution: BC should not distribute payment amount that have been disbursed \(BC-9783\)](#)

Issue – Reversing a DirectBillPayment is causing T-account Unapplied to become negative. This can happen when another direct bill payment on the same account gets modified.

Workaround – Create a recapture charge when unapplied goes below zero.

[Changing Billing Method From AB to DB for earning on First Payment Received For Billed Statement Is Earning Commission \(BC-10476\)](#)

Issue – Changing the billing method on a policy period from agency bill to direct bill with a producer that earns commission on First Payment Received will instead earn the commission when invoiced. Commission should not be earned until a payment is received.

Workaround – As part of transferring a policy period from agency bill to direct bill, add code to the policy period preupdate rule to reverse the earning transaction if the total amount paid on the policy period is zero. This will make the earning go back to zero for the policy period. The commission payable batch process will not identify these charge commissions (since their earning criteria will not match the selection criteria). When a real payment is posted to the policy, commission earning will be triggered by that event and be earned correctly.

[PolicyNumberLongDenorm not updated in upgrade \(BC-10742\)](#)

Issue – Prior to 3.0, BillingCenter used 0-based term numbers for policies. Starting with 3.0, BillingCenter uses 1-based term numbers for policies to be consistent with PolicyCenter term numbering. If you are not upgrading from a version prior to 3.0 this issue does not affect you.

Because the `PolicyNumberLong` field (and the corresponding `PolicyNumberLongDenorm` field) can be a legal reference, BillingCenter does not change the value during upgrade.

The upgrade does renumber the `TermNumber` from being 0-based to being 1-based, since the `TermNumber` is not normally used as a legal reference. This behavior is to avoid a gap between term numbers. Otherwise, the terms would have a gap. For example, instead of `TermNumber` 0, 1, <upgrade> 3, 4, the upgrade renbers the first two to make it 1, 2, 3, 4.

However, because the upgrade does not adjust `PolicyNumberLong` and `PolicyNumberLongDenorm`, there will be a gap in the `PolicyNumberLong` (for example 123456-0, 123456-1, <upgrade> 123456-3, 123456-4,. skipping any 123456-2 `PolicyNumberLong`).

Workaround – If you do not need to preserve your policy numbers after upgrade and you want your `PolicyNumberLong` and `PolicyNumberLongDenorm` values to match the new post-upgrade `TermNumber`, contact Guidewire Customer Support for information on how to update these fields.

IMPORTANT: The `PolicyNumberLong` and `PolicyNumberLongDenorm` columns must be kept in sync.

[UpgradeCommission batch process is throwing an error when a reversal or payment plan refactor happens after a point in time transfer \(BC-10917\)](#)

Issue – Under specific circumstances where a point in time transfer occurs, then a reversal or payment plan refactor occurs, then the BillingCenter database is upgraded to version 7.0.3, an error can later occur when running the `UpgradeCommission` batch process.

Workaround – Contact Guidewire Customer Support for assistance if you encounter this issue.

[Reversing Write-Off Commissions and Write-Off Commission Recovery in the UI \(BC-7184\)](#)

Issue – The BillingCenter user interface does not provide a way reverse Write-Off Commissions and Write-Off Commission Recovery.

Workaround – Clone the Reverse button under the Payments tab and reverse `CmsnPayableReduction` and other transactions using the following action: `CurrentLocation.startEditing()`; `producerTransaction.reverse()`; and `CurrentLocation.commit()`.

[Adding an existing contact following a new contact, causes the added contact not to display in the ListView on New Account and New Producer \(BC-881\)](#)

Issue – Attempting to add an existing contact following a new contact causes the added contact to not display properly in the list view on the Add Contact wizard on the New Account and New Producer screens.

Workaround – Click on a column heading and the sort cause the screen to refresh and display the contacts correctly.

[Rewrite BI with the same policymumber as the original policy is not incrementing the TermNumber and throws DBDuplicateKeyException \(BC-2961\)](#)

Issue – Sending a rewrite billing instruction without specifying the term number will throw a `DBDuplicateKeyException` if term number “1” already exists for the policy. If you omit the term number, BillingCenter uses the number “1” for term number. If a policy period with the term number “1” already exists then BillingCenter will throw an exception.

Workaround – Include a term number that is unique for the given policy number when sending a rewrite billing instruction.

[AGBL statement Total Payment Applied is not correct after change 1 of 2 producer policies payment schedule \(BC-2966\)](#)

Issue – The producer statement does not show the correct dollar amount after a change is made to one of the policy’s payment plans on a policy with two producers after the statement was billed and partially paid.

Workaround – Guidewire is aware of this issue.

Upgrade from BillingCenter 2.1.10 with an agency bill statement with a commission remainder invoice item may result in incorrect written off and net amounts for that agency bill cycle (BC-10756)

Issue – If you are upgrading from BillingCenter 2.1.10 and you have an agency bill statement with a commission remainder invoice item, the upgraded written off and net amounts on the agency bill cycle can be inaccurate. The agency bill statement will correctly reflect the commission remainder. Only the agency bill cycle has the incorrect (not adjusted by commission remainder write-off) amounts for net and written off amounts.

Workaround – Guidewire is aware of this issue.

Studio/Platform Issues

Problem with regen-java-api command and JAR files (PL-16351)

Issue – If you run the `BillingCenter/bin/gwpc regen-java-api` command, BillingCenter creates a `BillingCenter/java-api/lib` directory and puts JAR files for Java development in this directory. However, BillingCenter does not always perform this task correctly, resulting in some of the generated JAR files not being copied to this directory.

Workaround – For missing library files, copy or import into your Java development environment the missing library files from the following directory:

`BillingCenter/admin/lib`

Copy them into the following directory:

`BillingCenter/java-api/lib`

Renaming method or property throws ParseResultsException (PL-16633)

Issue – If you rename a property or a method or you change a method signature, and a workflow references that property or method in a Gosu field, BillingCenter throws `ParseResultsException`. This is the intended behavior.

Workaround – Restart the workflow engine. To do so:

1. Log into BillingCenter using an administrative account.
2. Access Internal Tools → Reload.
3. Click Reload Workflow Engine.

In some languages, web browsers render column headers of list views improperly (PL-18027)

Issue – In some languages, web browsers render some column headers of list views improperly if their column widths are specified too narrowly in their PCF definitions. For example, sometimes a numeric column is specified with a variable width of 1%. This narrow setting forces the browser to render the column too narrowly for the text of the translated column heading.

Workaround – Edit the PCF file that defines the column and clear the value from the width property. Without a specified value for the column width, browsers render the column widely enough to display the full text of the translated column heading.



chapter 23

Guidewire BillingCenter 7.0.5 Release Notes

Release 7.0.5

Release Notes Update: 22-July-2014

IMPORTANT These release notes replace the release notes that were included in the official product release. Please disregard the earlier version of the release notes.

Document Revision: 02-July-2015

Copyright © 2001-2015 Guidewire Software, Inc. All rights reserved.

Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire Reinsurance Management, Guidewire ContactManager, Guidewire Vendor Data Management, Guidewire Client Data Management, Guidewire Rating Management, Guidewire InsuranceSuite, Guidewire ContactCenter, Guidewire Studio, Guidewire Product Designer, Guidewire Live, Guidewire DataHub, Guidewire InfoCenter, Guidewire Standard Reporting, Guidewire ExampleCenter, Guidewire Account Manager Portal, Guidewire Claim Portal, Guidewire Policyholder Portal, Gosu, Deliver Insurance Your Way, and the Guidewire logo are trademarks, service marks, or registered trademarks of Guidewire Software, Inc. in the United States and/or other countries.

All other trademarks are the property of their respective owners.

This material is confidential and proprietary to Guidewire and subject to the confidentiality terms in the applicable license agreement and/or separate nondisclosure agreement.

Guidewire products are protected by one or more United States patents.

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release

- Support
- Issues and Major Changes
- Improvements and General Issues
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire BillingCenter.

Version Number

This release of Guidewire BillingCenter is 7.0.5.

- If you are integrating with Guidewire PolicyCenter, this version of BillingCenter requires PolicyCenter 7.0.4 or later.
- If you are integrating with Guidewire ContactManager, this version of BillingCenter requires ContactManager 7.0.0 or later. Guidewire recommends that you use the latest release available, which is currently ContactManager 7.0.5.

Installing This Release

IMPORTANT If you use an Oracle database, this release requires Oracle patch 12813641.

Refer to the following:

- The *BillingCenter Installation Guide* for general installation information
- Prior BillingCenter release notes for any versions that you have skipped

Support

For assistance, visit the Guidewire Resource Portal – <http://guidewire.custhelp.com>

Issues and Major Changes

This section contains issues and major changes that may affect your installation.

- Upgrade Support
- Base PCF File Changes
- Changes in This Release Provided in Upgrade Diff Report

Upgrade Support

- Upgrading from BillingCenter 2.1.9 or newer versions is supported in this release.
- This release includes multiple enhancements that will improve performance. As a result of some of these performance improvements, upgrade of large BillingCenter databases (one million or more policies per year) will take longer than upgrading to earlier maintenance releases. For more information see BC-12707 under BillingCenter Known Issues

- The following information is not yet included in the BillingCenter Upgrade Guide:
In cases where the definition of an Entity has been changed from being retireable to being non-retirable, BillingCenter 7.0.3 and later versions will not automatically remove data rows that had been retired. Instead, before upgrading to BillingCenter 7.0.3 or later versions, a new version check, `MigrateTableFromRetireableToEditableVersionCheck`, will run. This version check will log an error if any retired rows are found. If any such errors are logged, the retired rows must be examined and removed or restored before upgrade can continue.

Base PCF File Changes

All links below require the `readme_files` directory on your local disk.

BillingCenter release 7.0.4 to 7.0.5

- To view a report of the changes in the base PCF files in the `modules/bc` directory, [click here](#).
- To view a report of the changes in the base PCF files in the `modules/p1` directory, [click here](#).

Changes in This Release Provided in Upgrade Diff Report

Guidewire provides a report detailing certain differences between the current release and your prior release. This report describes changes in display keys, entities, typelists, and the Gosu API. To obtain your custom Upgrade Diff Report, visit the Guidewire Resource Portal.

Improvements and General Issues

BillingCenter Improvements and General Issues

The following are the improvements added to and the general issues corrected in this release:

ID	Description
Agency Bill	
BC-11929	<p>The Agency Bill new promise wizard has been updated to improve performance and to be consistent with the Agency Bill new payment wizard.</p> <p>As a result of this change, overpaid statements will now be displayed in the promise popup page.</p>
BC-12170	The Agency Bill Exceptions page was updated to improve performance.

ID	Description
BC-12177	<p>Fixed a performance problem on the producer agency bill open item and agency bill statement detail pages.</p> <p>The AgencyBillStatementView class has changed in two ways:</p> <ol style="list-style-type: none"> 1) The properties for getting totals relating to previous snapshot items (such as TotalGrossAmountForPreviousSnapshotItemsOnly, TotalCommissionAmountForPreviousSnapshotItemsOnly) now use a query to find these totals, rather than reading in many thousands of entities and calculating the totals in memory. Using a query assumes that the snapshot items are all committed to the database and do not change during the lifetime of the AgencyBillStatementView object. This is true in the base configuration of the pages that use the AgencyBillStatementView class, but it is possible that your configuration code does not follow this assumption. 2) The invoice items displayed on the statement detail and open items pages are now lazy loaded. If there are hundreds of pages of results to display BillingCenter previously read all the invoice items into memory immediately. BillingCenter now only loads them as needed. Therefor if you only display the first page of results then only those invoice items will be loaded. This is a significant performance improvement when there are a very large number of pages to display, and should have little effect if there are only a few pages to display. <p>However, if your configuration code iterates all the invoice items, this change will make that iteration much slower (because the invoice items are no longer all pre-loaded).</p>
BC-12194	<p>Fixed a problem where the agency bill open items page would fail to display correctly if there were any open invoice items that did not belong to a policy. (This is an unusual situation, most likely to be caused by explicitly assigning an item such as an account late fee to a producer). Such items are no longer displayed on the agency bill open items page.</p>
BC-12195	<p>Fixed a problem where the agency bill open items page was incorrectly displaying some settled invoice items.</p> <p>The most common case where this problem occurred was commission remainder items which were fully written off. These were still appearing on the open items page, which is incorrect.</p> <p>The page has now been updated so it will only display items that are not settled, as calculated by the InvoiceItem.Settled property.</p>
BC-12262	<p>Modified the Producer Payments page so it will perform better when displaying large payments.</p> <p>The original page was split into three separate pages: AgencyBillExecutedPayments.pcf, AgencyBillExecutedCreditDistributions.pcf and AgencyBillSavedPayments.pcf.</p> <p>These pages make up a location group called AgencyBillPayments.pcf. If you have modified the original ProducerPayments.pcf page in your configuration, you must manually merge those changes into the new pages. The three card views in ProducerPayments.pcf correspond to the three new pages.</p>
BC-12284	<p>Improved the performance of the Producer Transactions page.</p> <p>As part of this performance improvement, a drop down list for filtering the transactions by policy was removed.</p> <p>The existing Policy Starts With filter can be used instead.</p>
BC-12393	<p>The performance of the agency bill edit payment wizard has been improved. The performance improvement relies on the assumption that you will rarely, if ever, manually edit a large number of the thousands of items on a large agency bill payment. Previously BillingCenter performed many operations on all the payment items, whether or not they were altered by the user. Now, whenever possible, we try to only do significant work on the altered items.</p> <p>There is one change that may affect your configuration code. Before this change, modifying a payment resulted in a call to the IDelinquencyProcessExtensions.onChargesPaid plugin method for all possibly delinquent accounts associated with any charge on the payment. BillingCenter will now call this method only for accounts associated with charges where an item has actually been modified.</p>
BC-12394	<p>Improved performance of policy sorting in the Agency Bill payment wizard.</p>

ID	Description
BC-12822	<p>Addressed a performance issue that occurred when adding invoice items to an agency bill payment with very large numbers of invoice items due to the way the search criteria was cached.</p> <p>A Gosu object, <code>gw.search.InvoiceItemSearchCriteria</code>, replaces the non-persistent entity, <code>entity.InvoiceItemSearchCriteria</code>. The existing non-persistent entity has been deprecated but not removed.</p> <p>If this performance issue does not affect you, you can continue to use the entity version of the search criteria by not merging the changes in <code>InvoiceItemSearchPanelSet</code> and replacing references to <code>gw.search.InvoiceItemSearchCriteria</code> with the original entity, <code>InvoiceItemSearchCriteria</code>.</p> <p>The entity search criteria, <code>entity.InvoiceItemSearchCriteria</code>, is deprecated.</p>
BC-12888, BC-13127	<p>BillingCenter 7.0.3 introduced a restriction for agency bill payments that did not allow a negative commission or gross amount to be applied to a positive invoice item, or a positive commission or gross amount to be applied to a negative invoice item. This restriction prevented the valid use case of a producer returning commission that was overpaid, or making an offsetting negative payment to correct an earlier overpayment.</p> <p>Removing this restriction also restores the payment plan refactor behavior in BillingCenter 7.0.2 and earlier. Specifically, a payment plan refactor that occurs after an agency bill payment of the type described above may result in money remaining in Producer Unapplied instead of being reapplied to invoice items.</p>
BC-12655	<p>The Statement Billed batch process was sometimes producing incorrect denorm values related to commission (such as net amount) on agency bill statements. This problem has been fixed.</p> <p>The problem first occurred in BillingCenter 7.0.3.</p> <p>If you were using agency bill on BillingCenter 7.0.3 or BillingCenter 7.0.4, contact Guidewire Customer Support for assistance with finding and fixing possible incorrect data.</p>
Charge Handling	
BC-11886	<p>Fixed an issue where a race condition in the code could lead to the <code>Account.ChargeHeld</code> bit being set incorrectly.</p> <p>This condition could occur if multiple charge holds were removed in parallel, by different threads. In such a case it was possible for the account <code>ChargeHeld</code> bit to be left at 1 (true) even though it should be 0 (false).</p> <p>The fix prevents this race condition but does not correct any existing <code>ChargeHeld</code> bits that are incorrectly set.</p> <p>To see if your database has incorrectly set <code>ChargeHeld</code> bits, run the consistency checks on just the <code>bc_account</code> table (this can be done from <code>ServerTools -> Info Pages -> Consistency Checks</code>).</p> <p>If any bits are set incorrectly the consistency check results should indicate which accounts are affected. Contact Guidewire Customer Support for assistance in fixing this data.</p>
Command Line Tools	
BC-12936	<p>With this release, the amount of memory required to run some command line tools has increased. The default memory settings cover most cases, but some circumstances may require larger settings than base configuration provides. If a command line tool has insufficient memory to run, change the memory settings in the <code>memory.properties</code> file. For more information, see “Tuning Command Line Tool Memory Settings” in the <i>Installation Guide</i>.</p>
Commission	
BC-11643	<p>Calling the <code>InvoiceItem.remove()</code> method will now throw an <code>UnsupportedOperationException</code> if that item has been billed, has earned commission, or has had commission written off against it. This is to prevent data corruption that was not previously safeguarded against. A new method is available in Gosu, <code>createAndPlaceOffset()</code>, which reverses that invoice item instead of removing it.</p> <p>If you have configuration code that calls <code>remove()</code> on <code>InvoiceItem</code>, you should update your code to first call <code>canRemove()</code>. If <code>canRemove()</code> returns <code>false</code>, call <code>createAndPlaceOffset()</code> instead.</p>
BC-11861	<p>Optimized the way BillingCenter updates the statement column on producer contexts when constructing a new <code>StatementInvoice</code> in the Producer Payment work queue.</p> <p>The only impact is that the new mechanism BillingCenter uses to update the producer contexts will no longer fire any events. However the <code>ProducerContext</code> entity is already marked as ignore for events.</p>

ID	Description
BC-12039	<p>In BillingCenter 7.0.4 the log level of a debugging message in the Commission Payable work queue was incorrectly increased so that the message was included at the INFO log level.</p> <p>This has been corrected and in BillingCenter 7.0.5 the log level of this message (CMSN Processing commission payable distributed worker) has been reverted back to debug level, as in BillingCenter 7.0.3 and earlier releases.</p>
BC-12107	<p>When a policy is switched from direct bill to agency bill with no reversal of payments the next producer statement generated contained incorrect commission amount under:</p> <ol style="list-style-type: none"> 1. Policy Activity section for the particular policy. 2. In the popup (ProducerStatementPopup) invoked from clicking on the policy number here for the drill down with any value chosen for the AggregationType. <p>These problems have been fixed. The Event displayed on the ProducerStatementPopup has also been updated to show the EventType along with the payable criteria when the event is connected to an earning transaction.</p>
BC-12547	<p>Beginning in BillingCenter 7.0.3 there was a problem related to assignment of items. The problem occurred if you had a direct bill producer was earning commission on a policy period based on the On Invoice Fully Paid commission earning criteria. If you fully paid the first Invoice and then used assignment of items to assign a single InvoiceItem to an agency bill producer and used the setting reversepayments = yes, then the CommissionReserve would not be adjusted as part of the payment reversal and therefore not transferred to the new ItemCommission.</p> <p>If you have been using BillingCenter 7.0.3 or BillingCenter 7.0.4 and the On Invoice Fully Paid commission earning criteria and transferring invoice items on those policies to agency producers, you should validate your database consistency checks. Contact Guidewire Customer Support for assistance if you find any problems.</p>

Product Configuration

BC-12257	<p>The ordering of batch processes in scheduler-config.xml was updated to use the generally recommended order.</p> <p>The new ordering in scheduler-config.xml represents the recommended ordering and does not imply a problem if you use a different order in scheduler-config.xml.</p>
----------	---

Consistency Checks

BC-12213	When an account level charge is paid by another account, a database consistency check was returning false positives. This error has been fixed.
BC-12360	For very large databases it can be very expensive to run the consistency checks. In our performance testing Guidewire found that some of the queries used in the consistency checks used a large amount of temporary space on Oracle. We have provided configuration parameters that allow some of these queries to be split into multiple smaller queries, each of which check only a portion of the database. These parameters should only be set if you have a very large database. The new configuration parameters are: MaxTransactionsInConsistencyCheck, default value 0 (meaning don't split) MaxTAccountsInConsistencyCheck default value 0 (meaning don't split)

Database Upgrade

BC-12011	<p>Fixed a problem with payment instrument upgrade. Now BillingCenter does the following:</p> <ol style="list-style-type: none"> 1) For Accounts and Producers, upgrade will convert a PaymentMethod to a PaymentInstrument using the PaymentMethod type. 2) For Accounts and Producers, default PaymentInstrument is set according to the pre-upgrade PaymentMethod.
BC-12639	Upgrading from BillingCenter 2.1 was incorrectly upgrading AcctNegativeWriteoffs when upgrading PolicyNegativeWriteoffs. This problem has been fixed.
BC-12726	Improved performance of the upgrade trigger, DisbursementConvertPaymentMethodDetailsVersionTrigger.

ID	Description
BC-12776	<p>When upgrading from BillingCenter 2.1, zero dollar payments were resulting in a null PaymentInstrument in the post upgrade database.</p> <p>The problem has been fixed and now the payment instrument correctly points to the singleton unapplied PaymentInstrument.</p>
Funds Tracking	
BC-12185	<p>The FundsAllotment batch process was failing with an error.</p> <p>This problem has been fixed and the error no longer occurs.</p>
BC-12867	Fixed a problem that sometimes caused an error when the FundsTracking batch process was run.
Internationalization	
BC-11646	Internationalized the formatting of currency amounts in NewCommissionPaymentDV.pcf.
BC-11647	Internationalized the formatting of currency amounts in the NewNegativeWriteoffDetailsDV.pcf.
BC-11648	Internationalized the formatting of currency amounts in NewNegativeWriteoffWizardConfirmationStep-Screen.pcf.
BC-11649	Internationalized the formatting of currency amounts in the NewNegativeWriteoffWizardConfirmation-StepScreen.pcf.
BC-11650	Internationalized the formatting of currency amounts in the CreditDetailsConfirmationScreen.pcf.
BC-11870	Charge#getTAccountOwnerTypeAndDisplayName() was not using the display name for the TAccountOwner. That was fixed and display names added for entities that are TAccountOwners (such as displaykey.entity.Account).
Invoicing	
BC-10778	Fixed a problem where the Invoice batch process would fail with an Illegal State Exception error if the Invoice batch process was restarted and the prior execution made invoices billed and then failed.
BC-12278	InvoiceAssembler.withContext() is now available and InvoiceAssembler was moved from com.guidewire.bc.domain.invoice to gw.api.domain.invoice
BC-12476	Fixed a problem where the invoice batch process would fail with an error if restarted and the batch made invoices due and committed those changes before being restarted.
PolicyCenter Integration	
BC-9690	<p>BillingCenter is now copying the original policy contact roles to new policy periods.</p> <p>If you do not want this behavior in your configuration, the policy contact roles may be deleted in Gosu after the renewal is created.</p>
Policy Closure	
BC-9090	Commission expense roll up transactions will now no longer be created if the BillingCenter config parameter CreateRollupTxnsUponPolicyClosure is set to false.
Performance	
BC-10665	Improved the performance of the producer total commission expense consistency check.
BC-11638	<p>There was a problem where BillingCenter could create ReserveCmsnEarned or CommissionReduction transactions and did not detect them when determining if an InvoiceItem could be removed. This allowed for the creation of earning transactions that point to a retired InvoiceItem.</p> <p>This problem has been fixed.</p>
BC-11899	Improved the performance of payment plan refactoring.
BC-11985	<p>The InvoiceDue work queue was updated to process invoices with a status of CarriedForward or WrittenOff only if those invoices have items that do not have a ChargeDue transaction. As a result, the InvoiceDueWorkQueue will process invoices with CarriedForward or WrittenOff statuses only once.</p> <p>Previously invoices with CarriedForward or WrittenOff statuses were processed each time the batch process ran. This would cause money to be distributed from the Unapplied T-account of the related accounts. Delinquency processing was also done for those accounts and policies.</p>
BC-12156	<p>The agency bill cycles page (ProducerAgencyBillCycles.pcf) has been updated to improve performance.</p> <p>This change included moving inline Gosu filters from the agency bill cycles page to a new static Gosu class, ProducerAgencyBillCycleFilters.gs.</p>

ID	Description
BC-12240	<p>The following changes were made to improve the performance of agency bill exception handling during payment reversals.</p> <ul style="list-style-type: none"> - Added an array of ProdWriteoffContext objects to AgencyCycleDist and an array of ChrgePaidFromProdCtx objects to AgencyPaymentItem. - Created a new Interface named InvoiceItemExceptionContext and two implementations of the interface (InvoiceItemExceptionContextBase and ReverseAllDistItemsExceptionContext). - Modified the signature of the methods in the IIInvoiceItem plugin interface to take in the above mentioned InvoiceItemExceptionContext instead of an InvoiceItemExceptionEvent. - Added the following alert message to the payment reversal confirmation page (AgencyPaymentReversalConfirmationPopup.pcf) to warn users reversing large payments: "Warning: This payment contains over 10,000 invoice items. Reversing it may take 5-10 minutes".
BC-12273	Performance of a number of agency bill pages was improved by adding a denorm column to the item commission table. The new denorm column contains data from the policy commission table.
BC-12707	In order to allow partitioning of the LineItem table in Oracle databases, the uniqueness constraint that was enforced by a Unique Index on the LineItem table was removed.
BC-12992	Added an index to Account Disbursement to improve a slow query in the automatic disbursement batch process.
Payment Handling	
BC-12906	<p>The New Payment batch process has been updated. Before this change, the New Payment batch process would create work items for all undistributed direct bill payments whether or not any money was available to distribute.</p> <p>Now the New Payment batch process checks to see if the target Account Unapplied balance for each payment is positive. If not, there is no money available to distribute and therefore no work item is created for that payment.</p>
Security	
BC-12958	<p>Internal BillingCenter code included a validation that required a producer write-off have an amount no greater than the ProducerWriteoffThreshold value from the producer's agency bill plan.</p> <p>This validation was redundant because identical configurable validation existed in the agency bill payment wizard user interface and in the agency payment API. The internal BillingCenter validation code was removed because it was not configurable.</p> <p>Note: If you have added configuration code that executes agency bill payments with write-off amounts directly in your configuration code (without going through the BillingCenter user interface or API that has configurable validation checking the ProducerWriteoffThredhold) you must update your configuration to add this validation to your configuration code.</p>
Suspense Payments	
BC-13114	Modified suspense items can now be released without a null pointer exception error.
Staging Tables	
BC-9046	NegativeWriteoff objects can now be loaded from staging tables.
Write-offs	
BC-12210	<p>To improve performance of the Producer Write-Offs page, it has been modified to include a drop-down date filter which allows the user to constrain the set of write-offs displayed.</p> <p>The base configuration of the date filter displays only those write-offs which were created within the last 30 days by default. The filter also has options to choose the last 60 days, 90 days, 365 days, or to show all write-offs. You can configure the exact set of options you want in the date filter by modifying the new Gosu class ProducerWriteOffAgeRestriction.gs</p>

Platform Improvements and General Issues

Archiving

- PL-26929 Modified the `restoreSeveredTransactionOffsetOnsetOnsetLinks` method so it can be called from Gosu.

Cognos integration

- PL-25031 With this release, the LDAP component of the BillingCenter integration with the Cognos report server properly handles user names that contain certain special characters.

Command line tools

- PL-20946 With this release, the `regen-pcfmapping`, `regen-dictionary`, and `verify-types` tools have a `maxSPVInclusions` parameter. The parameter format on the command line is `-DmaxSPVInclusions=n`, where `n` is the maximum number of shared sections. This parameter limits the number of shared sections during second-pass PCF verification, which can run for extremely long periods of times in a configuration with many nested modal PCF widgets.
- PL-23702 `JAVA_OPT` parameters passed to a `CmdLineToolBase` subclasses are no longer printed in console output.
- PL-25997 With this release, the amount of memory required to run some command line tools has increased. The default memory settings cover most cases, but some circumstances may require larger settings than base configuration provides. If a command line tool has insufficient memory to run, change the memory settings in the `memory.properties` file. For more information, see “Tuning Command Line Tool Memory Settings” in the *Installation Guide*.

Configuration upgrade

- PL-18868 The Configuration Upgrade Tool moves typelist localizations into `typelist.properties` files. However the tool did not place the customized `typelist.properties` files into the configuration module.

Database instrumentation

- PL-23475 When generating database statistics, if a column existed in the database that was not defined in the data model, the statistics generation would fail. This issue has been resolved.

Database support

- PL-25889 In earlier releases, the Database Statistics batch process ran count queries on tables for which statistics were updated. With this release, the batch process obtains counts from statistics views to reduce the performance burden on the database.
- PL-27436 This release adds support for Oracle date interval partitioning. See “Configuring Oracle Date Interval Partitioning” in the *System Administration Guide*.

Database upgrade

PL-23686	<p>With this release, generating table statistics during upgrade is optional for Oracle databases. This change does not affect statistics generation on the Microsoft SQL Server and H2 development databases.</p> <p>You set this option in the <code>config.xml</code> file. In the <code><database></code> block, there is an <code><upgrade></code> block that contains configuration information for the overall database upgrade. There is a new attribute in the <code><upgrade></code> block, <code>updatestatistics</code> with default value <code>true</code>. If you set this attribute to <code>false</code>, statistics will not be updated for that database during upgrade.</p> <p>If you do not update statistics during upgrade, you are then responsible for updating statistics by using the command-line batch process. Most of the time, the incremental statistics update is sufficient. For example:</p> <pre><code>maintenance_tools -password password -startprocess incrementaldbstats</code></pre> <p>If statistics are not updated during the upgrade, you see a warning that recommends that you run the database statistics batch process in incremental mode. Additionally, the UpgradeInfo page shows that statistics were not updated as part of the upgrade. This page also reports the runs of the statistics batch process, and incremental runs are shown.</p> <p>The UpgradeInfo page does not identify the following case: You ran another upgrade with <code>updatestatistics=true</code> since running a previous update with <code>updatestatistics=false</code>, but you did not update statistics first.</p> <p>When you click the Download button on the UpgradeInfo page, you get a more detailed UpgradeInfo report. The UpgradeInfo report shows the value of the <code>updatestatistics</code> attribute at the time of upgrade.</p> <p>Additionally, the UpgradeInfo report shows the update statistics SQL statements that were skipped as part of the upgrade. Normally, you do not need to consult this list, because running the database statistics batch process in full mode will cover these statements.</p> <p>Note: The incremental mode might not capture all cases that would have been run during an upgrade with <code>updatestatistics=true</code>.</p>
PL-23785	Altered the algorithm used to calculate column statistics for database upgrade.
PL-24051	With this release, the default configuration of the <code>createIndexInParallel</code> attribute in the <code><upgrade></code> block is set to <code>DEFAULT</code> instead of <code>1</code> . The value <code>DEFAULT</code> instructs Oracle to determine automatically the degree of parallelism to use. To turn off parallelism completely, set the <code>createIndexInParallel</code> attribute to <code>1</code> . The <code>createIndexInParallel</code> attribute is used only for Oracle databases.
PL-24204	Resolved an issue where failures occurred if you include the <code>sqlserverCreateIndexSortInTempDB</code> attribute in the <code><upgrade></code> element.
Document management	
PL-21883	This release introduces a document template descriptor attribute called <code>mailmergetype</code> , which optionally configures pagination of client-side Microsoft Word production. In the immediately previous release, by default, the pagination is <i>catalog pagination</i> . Catalog pagination correctly trims the extra blank page at the end. However, catalog pagination does not permit template substitution in headers and footers. The default behavior did not change in this release. However, if your template descriptor file includes the optional attribute <code>mailmergetype</code> in the <code><DocumentTemplateDescriptor></code> element set to any value other than <code>catalog</code> , the behavior changes. The non-catalog behavior is that a blank page is added to the end of the file. However, headers and footers support template substitution.
PL-24790	This release resolves an issue with handling large images in RTF templates that resulted in a <code>ClassFormatError</code> exception.
PL-25361	With this release, you can use Word document templates that have no form fields without errors.
Email	
PL-16353	If the filename of an email attachment contained Japanese characters, the filename was not retained.
Entities/metadata	
PL-22742	This release resolves an issue with search denormalization columns not being updated when only the source entity is changed, and the source column is itself a linguistic search denormalization column, resulting in a consistency check error on <code>ClaimContact</code> denormalization columns.
PL-24004	In previous releases, field validators prevented the restoration of archived entity graphs if an archived field contained data that violated field validation for its datatype. With this release, BillingCenter turns off field validation during the restoration of entity graphs from the archiving data store.

PL-24640 This release resolves an issue in Gosu that led to an infinite loop and a stack overflow exception when constructing an entity type. One side effect of this change is that on a Gosu class or interface, you cannot have two methods with signatures that differ only by entity where the entity types (versionable, retireable, and so on) are the same.

For example, assume that you have two entities, A and B, that are both retireable. The following example code does not compile:

```
class Foo {
    function foo(a: A) {}
    function foo(b: B) {} // Entity type is same as previous method, so signatures
                         // are considered equivalent
}
```

You can change the preceding code so it compiles by renaming one of the methods to differ from the other. If entities A and B were of different types, then the preceding code would compile without changing method names.

External entities

PL-20534 This release resolves an issue where adding a method to an entity enhancement showed as deprecated on Java API classes, the files generated by the regen-java-api command.

Geocoding/proximity search

PL-22475 Turn-by-turn driving directions were not provided in the native language.

Gosu

PL-24095 This release resolves an issue in which casting errors were thrown when Gosu code called a Java method that expected a specific entity, such as Document, as a parameter. The system would throw an error similar to the following:

```
gw.lang.parser.EvaluationException:
com.guidewire.commons.metadata.proxy._generated.impl.Document cannot
be cast to com.guidewire.pc.external.entity.Document
```

This error was caused by the way in which Guidewire Java code was making a reflective method call to Gosu and then handling external entity types. Those reflective method calls no longer require casting of the external entity types.

PL-24177 This release resolves an issue that caused the exists method to throw an exception when the where clause includes a Guidewire boxed Boolean instead of a Java primitive boolean.

GX Tools

PL-25738 This release Improves the XML serialization performance of data generated by the GX model tool whenever sorting is enabled.

Integration

PL-23132 Problems occurred when starting a listening thread on JMS if the application server was the latest version of WebSphere. With this release, you can use a listening thread with a startable plugin on any version of WebSphere or WebLogic.

Localization

PL-12799 In previous releases, when you exported display keys from Studio by using the Export Translation File command, occurrences of # were converted to \#. Beginning with this release, they are exported unchanged as #.

PL-24462 In previous releases, BillingCenter incorrectly handled Imperial dates that equate to Gregorian Dates in the year 1950, as well as other ranges of old dates. The issue occurred because American military bases used daylight saving time in Japan beginning in 1948, which differed from timekeeping used by the general public. With this release, BillingCenter handles old Imperial dates correctly.

Messaging

PL-26571 This release introduces a new a new attribute, `multithreadnonsafeorderedmessages`, in the file `messaging-config.xml`. For each destination in `messaging-config.xml`, you can add the attribute to indicates that the absolute order of non-safe-ordered messages does not need to be maintained. If you set that attribute to true for a destination, BillingCenter uses multiple messaging threads instead of a single thread to send non-safe-ordered messages in parallel. Without the attribute or when set to false, BillingCenter uses a single message thread for the destination and sends non-safe-ordered messages sequentially.

Note: The Messaging editor in Studio does not let you view or edit this new attribute of a destination.

Profiling

PL-25624	Fixed an issue where custom batch processes that handled large amounts of data caused errors in Guidewire Profiler.
----------	---

Query builder APIs

PL-12947	This release resolves an issue with improper joins whenever doing column selects for EffDated tables by disallowing them in view entities.
PL-23688	Javadoc has been updated to reflect the difference between the cast and join methods with respect to entity subtypes.

Server tools pages

PL-23991	<p>This release introduces a Configuration Info page that shows current values of system parameters and allows downloading the configuration dump as a ZIP archive.</p> <p>The ZIP archive includes copies of the main configuration files, as well as dumps of the in-memory state of the configuration on the server. The in-memory state reports the state of certain types of configuration settings that change after the server starts via web service and JMX APIs.</p> <p>File system configuration files:</p> <ul style="list-style-type: none"> • config.xml • config-override.xml (there might be several) • work-queue.xml • scheduler-config.xml • messaging-config.xml <p>In-memory configuration files:</p> <ul style="list-style-type: none"> • config.xml • work-queue.xml
----------	---

PL-24010	This release introduces the Download Raw Report button on the Work Queue Info page. Click the button to generate a downloadable report in CSV format. The report contains time-sliced raw data from the ProcessHistory and InstrumentedWorkerTask tables for analysis with third party tools such as Microsoft Excel.
PL-24011	The Work Queue Info page has a Download button that generates a downloadable report. This release introduces a new page in the report, Work Queue Runs. The Work Queue Runs page summarizes each work queue run over the specified time range.

Utilities

PL-16503	This release resolves an issue in the class gw.api.print.ListViewPrintOptionPopupAction in the printListViewOnlyWithOptions function where some objects did not print when exported to CSV files.
PL-25064	With this release, the regen-dictionary and verify-types commands now prints INFO log messages to show progress every five minutes. Sometimes the regen-dictionary and verify-types processes take a long time because they use modal PCFs. The progress messages help you determine where problems occur.

Web - UI/runtime

PL-24770	Users could not make a selection before the screen refreshed. Fixed interaction of autocomplete with postOnChange, when used in conjunction with a contains.
PL-25291	This release resolves a JavaScript issue that did not allow users to copy-and-paste into fields with an input mask.
PL-25666	This release resolves an issue where printing or exporting a List View to Excel/CSV did not populate the values of RangeCell in the output.

Web services

PL-24615	This release introduces a parameter in config.xml called LoadSoapServicesOnStartup that affects RPCE Web Service publishing. This parameter existed in older versions of Guidewire products: ClaimCenter 5, PolicyCenter 3, BillingCenter 2. If present and set to true, for RPCE web services, instantiation of all the implementation classes and registering of the WSDL happens on server startup. Otherwise, initialization of all implementation classes happens on the first request to any service. If you use clusters of servers, you can set this parameter differently on each server. For example, if you have only one server handling web service requests, if you set this parameter only on that one server, startup time on the other servers is unaffected. This parameter only affects RPCE web services, not WS-I web services. WS-I web services do not construct objects until detecting a request for that specific web service.
PL-26973	With this release, request-scoped and session-scoped variables that use the RequestVar and SessionVar classes work within RPCE web service implementation classes. This improvement does not affect web services running locally, which are supported only for testing. In other words, unit tests that run web services from the soap.local namespace cannot use RequestVar and SessionVar classes.
PL-26857	For WS-I web services, this release changes one server behavior and one client behavior. On the server, this release changes the behavior of the annotation @WsiCheckDuplicateExternalTransaction. If you add this annotation to your implementation class, the SOAP header <transaction_id> is required in each request, rather than optional. On the client with code in Gosu, this release introduces a new utility method that you can call to set the transaction ID for the next call. On the API object, call: <code>api.initializeExternalTransactionIdForNextUse(transactionIDString).</code> After the call completes, the application removes the transaction ID from that API object to prevent accidental reuse.
PL-26588	For WS-I web services, you can log the XML request and responses for one or more web services. If you set a debug-level logger entry for the class name of a WS-I web service, the application will log the request and response to that appender. Set the logging level to DEBUG in your logging configuration file.
PL-27035	For WS-I web services, this release resolves an issue with reliable transfers of binary data using MTOM, in some cases.
Work queues	
PL-23027	This release enhances BulkInsertWorkQueue to allow creation of custom work item types, not just StandardWorkItem.
PL-23427	With this release, the signature of the IBatchCompletedNotification.complete method changes from: <code>public void completed(BatchProcessType batch);</code> to: <code>public void completed(ProcessHistory batch, int workItemsFailed);</code> The new signature allows passing the number of failed work items. In earlier releases, BillingCenter did not call the complete method if the queue had work items that failed. With this release, BillingCenter calls the complete method if the queue has nothing but work items that failed.
PL-24012	This release introduces an Item Processing Time column on the Work Queue Info report and adds the to the Worker(s) Summary by Host and Recent Batches tables. The column reports work item processing time. This information enables calculating the efficiency of a work queue as the ratio of work item processing time to total active time of a worker. Time that an active worker does not spend on processing work items is spent on interaction with the work queue infrastructure.
PL-24325	This release introduces the optional attribute checkInAfterError to the <work-queue> element. If checkInAfterError is set to true for a work queue, BillingCenter attempts to process failed work items again immediately after failure. The default value of checkInAfterError is false.
XML element and XSD types	
PL-21636	In previous releases, if data exported to a GX model contained invalid XML characters, as specified in the XML specification, an exception was thrown. With this release, invalid XML characters are removed from the exported data automatically.
PL-27090	With this release, XML/XSD types is Gosu support use of circular xs:include references.

Known Issues and Limitations

This section describes known issues with this release of Guidewire BillingCenter.

- BillingCenter Known Issues
- Studio/Platform Issues

BillingCenter Known Issues

[Performance improvements cause upgrade to take longer than usual in a maintenance release \(BC-12707\)](#)

Issue – BillingCenter 7.0.4 and BillingCenter 7.0.5 include multiple performance enhancements that will result in better performance, especially for large databases. As a result of these changes, large volume customers will notice that upgrade takes longer than most maintenance releases. The one-time cost to large volume customers of a longer upgrade is necessary to achieve the benefit to those same customers of permanent performance improvements.

Guidewire internal tests using a database containing one million policies with one year of data took an hour on SQL Server. There should be minimal impact to customers with less than one million policies.

Workaround – Guidewire recommends performing a test upgrade in an environment that simulates your production environment, including using similar hardware and a similar size database.

For assistance in estimating how long upgrade will take please contact Guidewire Customer Support.

[Delinquency: Delinquency can be started on an account, even though there is no amount due, when negative charges are present \(BC-3746\)](#)

Issue – Delinquency can be started on an account when no amount is due if a negative charge on a policy is present.

Workaround – Invoke the `doDueTransaction()` method on the negative invoice item. This will execute a `ChargeDue` transaction for that invoice item which will prevent the problem.

Note that you can execute `invoiceItem.doDueTransaction()` method on a negative invoice item even if it is on an invoice that is in billed state not yet due. Later, when the invoice is made due, BillingCenter will just skip the negative invoice (because `invoiceItem.doDueTransaction()` already forced that invoice item to be due).

[PolicyCenter Integration: BC Renewal thread fails with exception on issue after removal of Alt Billing Account from Renewal in PC. \(BC-9540\)](#)

Issue – If you are integrated with PolicyCenter, you will encounter an error if you renew a policy that had an alternate billing account and you remove the alternate billing account in PolicyCenter prior to committing renewal.

Workaround – Guidewire is aware of this issue.

[Contact Manager: BC does not show updated Primary Address after CM contact Merge \(BC-9315\)](#)

Issue – If you are using BillingCenter with Contact Manager, there is a problem with merging contacts with different addresses. The primary address of the merged contact is not shown in BillingCenter.

Workaround – Guidewire is aware of this issue.

Commission: Commission override after a commission write-off causes write-off to be reserved (BC-9536)

Issue – Several methods of the BillingAPI WS-I webservice, such as `issuePolicyPeriod()` and `cancelPolicyperiod()`, indirectly call `BillingInstructionUtil.executeAndCommit()` which commits the current bundle.

These methods also directly call `tryCatch()`, which commits a new bundle after executing the entire block of code inside of `tryCatch()`. As a result, the code tries to commit the same bundle twice.

Here are two ways you can encounter this problem:

1. If you try to create an activity within the block of code passed to `tryCatch()` after the call to execute the billing instruction, BillingCenter will throw a `DBA1readyExecutedException`.

For example:

```
function issuePolicyPeriod(issuePolicyInfo : IssuePolicyInfo, transactionId : String) : String{
    require(issuePolicyInfo, "issuePolicyInfo")
    var publicID = tryCatch( \ bundle -> {
        var pid= issuePolicyInfo.executeIssuanceBI()
        // code *here* to create a new activity
        return pid
    }, transactionId)
    return publicID
}
```

2. If you used `CreateTroubleTicketHelper` to create a trouble ticket in a billing instruction preupdate rule, you would again get a `DBA1readyExecutedException`. The following is an example of a billing instruction preupdate rule that would trigger the error:

```
// create trouble ticket with a delq hold until second invoice
var troubleTicketHelper = new CreateTroubleTicketHelper()
var ticket = troubleTicketHelper.createTroubleTicket((billingInstruction as
    PlcyBillingInstruction).PolicyPeriod)
ticket.Title = "Delinquency hold on issuance"
ticket.DetailedDescription = ticket.Title
ticket.Priority = "normal"
ticket.TicketType = "automatic"
ticket.Hold.setAppliedToHoldType(HoldType.TC_DELINQUENCY, true)
ticket.Hold.setReleaseDate(HoldType.TC_DELINQUENCY, gw.util.GwBaseDateEnhancement.CurrentDate)
```

Workaround –

1. For this case, in the block of code passed to `tryCatch()`, do not create or update entities after the call to execute the billing instruction. For example:

```
function issuePolicyPeriod(issuePolicyInfo : IssuePolicyInfo, transactionId : String) : String{
    require(issuePolicyInfo, "issuePolicyInfo")
    var publicID = tryCatch( \ bundle -> {
        // code *here* to create a new activity
        var pid= issuePolicyInfo.executeIssuanceBI()
        return pid
    }, transactionId)
    return publicID
}
```

2. For this case, instead of using `CreateTroubleTicketHelper` to create the trouble ticket, create related entities directly as in the following example in a PolicyPeriod enhancement:

```
var ticket = new TroubleTicket(this)
var ttJoinEntity = new TroubleTicketJoinEntity(ticket)
ticket.Title = ticketTitle
ticket.DetailedDescription = "Delinquency hold placed on Account with a " + ticket.Title
ticket.Priority = "normal"
ticket.TicketType = ticketType
ticket.Hold.setAppliedToHoldType( HoldType.TC_DELINQUENCY, true )
ticket.Hold.setReleaseDate(HoldType.TC_DELINQUENCY, releaseDate)
ticket.addToTroubleTicketJoinEntities(ttJoinEntity)
this.addToTroubleTicketJoinEntities(ttJoinEntity)
this.Account.addToTroubleTicketJoinEntities(ttJoinEntity)
ticket.Hold.checkForHoldAdditions()
```

[PolicyCenter Integration: BC Renewal thread fails with exception on issue after removal of Alt Billing Account from Renewal in PC \(BC-9540\)](#)

Issue – If you are integrated with PolicyCenter, you will encounter an error if you renew a policy that had an alternate billing account and you remove the alternate billing account in PolicyCenter prior to committing renewal.

Workaround – Guidewire is aware of this issue.

[PolicyCenter Integration On Renewal Offer, Suspense Payment Should Be Applied At Policy Period \(BC-9691\)](#)

Issue – When integrated with PolicyCenter, a suspense payment that is created with an offer number from a PolicyCenter renewal is getting applied to the owner account of the policy period. The suspense payment should be applied to the renewal policy period instead of the owner account.

Workaround – Guidewire is aware of this issue.

[Payments: Reversing Producer Payments is throwing NullPointerException when the payment is related to a retired invoice item \(BC-9749\)](#)

Issue – BillingCenter does not allow invoice items that have earned commission to be deleted. BillingCenter already prevents such items from being removed through the user interface. If an invoice item that has earned commission is deleted programmatically, errors will occur if the related payment item is reversed.

Workaround – Instead of deleting invoice items that have earned commission, set the invoice item amount to zero.

[Disbursement: Changing the Disbursement reason from Automatic causes it to be sent out despite a hold \(BC-9761\)](#)

Issue – The Disbursements batch process sends an disbursement if the disbursement reason is changed from 'Automatic' to something else, even if there is an active Trouble Ticket with a hold of type 'Disbursements' on the owning account. The BillingCenter user interface does not allow editing of automatic disbursements, but they can be edited through configuration. BillingCenter fails to account for this and the disbursement batch process treats a reason other than automatic to mean that it is a manual disbursement and so processes it, ignoring any holds.

Workaround – Use a custom extension to store more specific reasons for an automatic disbursement.

[Payment Distribution: BC should not distribute payment amount that have been disbursed \(BC-9783\)](#)

Issue – Reversing a DirectBillPayment is causing T-account Unapplied to become negative. This can happen when another direct bill payment on the same account gets modified.

Workaround – Create a recapture charge when unapplied goes below zero.

[Changing Billing Method From AB to DB for earning on First Payment Received For Billed Statement Is Earning Commission \(BC-10476\)](#)

Issue – Changing the billing method on a policy period from agency bill to direct bill with a producer that earns commission on First Payment Received will instead earn the commission when invoiced. Commission should not be earned until a payment is received.

Workaround – As part of transferring a policy period from agency bill to direct bill, add code to the policy period preupdate rule to reverse the earning transaction if the total amount paid on the policy period is zero. This will make the earning go back to zero for the policy period. The commission payable batch process will not identify these charge commissions (since their earning criteria will not match the selection criteria). When a real payment is posted to the policy, commission earning will be triggered by that event and be earned correctly.

[PolicyNumberLongDenorm not updated in upgrade \(BC-10742\)](#)

Issue – Prior to 3.0, BillingCenter used 0-based term numbers for policies. Starting with 3.0, BillingCenter uses 1-based term numbers for policies to be consistent with PolicyCenter term numbering. If you are not upgrading from a version prior to 3.0 this issue does not affect you.

Because the `PolicyNumberLong` field (and the corresponding `PolicyNumberLongDenorm` field) can be a legal reference, BillingCenter does not change the value during upgrade.

The upgrade does renumber the `TermNumber` from being 0-based to being 1-based, since the `TermNumber` is not normally used as a legal reference. This behavior is to avoid a gap between term numbers. Otherwise, the terms would have a gap. For example, instead of `TermNumber` 0, 1, <upgrade> 3, 4, the upgrade renames the first two to make it 1, 2, 3, 4.

However, because the upgrade does not adjust `PolicyNumberLong` and `PolicyNumberLongDenorm`, there will be a gap in the `PolicyNumberLong` (for example 123456-0, 123456-1, <upgrade> 123456-3, 123456-4,. skipping any 123456-2 `PolicyNumberLong`).

Workaround – If you do not need to preserve your policy numbers after upgrade and you want your `PolicyNumberLong` and `PolicyNumberLongDenorm` values to match the new post-upgrade `TermNumber`, contact Guidewire Customer Support for information on how to update these fields.

IMPORTANT: The `PolicyNumberLong` and `PolicyNumberLongDenorm` columns must be kept in sync.

[UpgradeCommission batch process is throwing an error when a reversal or payment plan refactor happens after a point in time transfer \(BC-10917\)](#)

Issue – Under specific circumstances where a point in time transfer occurs, then a reversal or payment plan refactor occurs, then the BillingCenter database is upgraded to version 7.0.3, an error can later occur when running the `UpgradeCommission` batch process.

Workaround – Contact Guidewire Customer Support for assistance if you encounter this issue.

[PolicyNumberLongDenorm not updated in upgrade \(BC-10742\)](#)

Issue – Prior to 3.0, BillingCenter used 0-based term numbers for policies. Starting with 3.0, BillingCenter uses 1-based term numbers for policies to be consistent with PolicyCenter term numbering. If you are not upgrading from a version prior to 3.0 this issue does not affect you.

Because the `PolicyNumberLong` field (and the corresponding `PolicyNumberLongDenorm` field) can be a legal reference, BillingCenter does not change the value during upgrade.

The upgrade does renumber the `TermNumber` from being 0-based to being 1-based, since the `TermNumber` is not normally used as a legal reference. This behavior is to avoid a gap between term numbers. Otherwise, the terms would have a gap. For example, instead of `TermNumber` 0, 1, <upgrade> 3, 4, the upgrade renames the first two to make it 1, 2, 3, 4.

However, because the upgrade does not adjust `PolicyNumberLong` and `PolicyNumberLongDenorm`, there will be a gap in the `PolicyNumberLong` (for example 123456-0, 123456-1, <upgrade> 123456-3, 123456-4,. skipping any 123456-2 `PolicyNumberLong`).

Workaround – If you do not need to preserve your policy numbers after upgrade and you want your PolicyNumberLong and PolicyNumberLongDenorm values to match the new post-upgrade TermNumber, contact Guidewire Customer Support for information on how to update these fields.

IMPORTANT The PolicyNumberLong and PolicyNumberLongDenorm columns must be kept in sync.

Should not be able to edit Payment Plan after it has been used by a policy (BC-1726)

Issue – In the base configuration, it is possible to edit a Payment Plan even after it has been used by a policy. This is incorrect. Only the expiration date should be updateable.

Workaround – Configure the payment plan screen to only allow the expiration date to be updated.

Reversing Write-Off Commissions and Write-Off Commission Recovery in the UI (BC-7184)

Issue – The BillingCenter user interface does not provide a way reverse Write-Off Commissions and Write-Off Commission Recovery.

Workaround – Clone the Reverse button under the Payments tab and reverse CmsnPayableReduction and other transactions using the following action: CurrentLocation.startEditing(); producerTransaction.reverse(); and CurrentLocation.commit().

Adding an existing contact following a new contact, causes the added contact not to display in the ListView on New Account and New Producer (BC-881)

Issue – Attempting to add an existing contact following a new contact causes the added contact to not display properly in the list view on the Add Contact wizard on the New Account and New Producer screens.

Workaround – Click on a column heading and the sort cause the screen to refresh and display the contacts correctly.

Rewrite BI with the same policymumber as the original policy is not incrementing the TermNumber and throws DBDuplicateKeyException (BC-2961)

Issue – Sending a rewrite billing instruction without specifying the term number will throw a DBDuplicateKeyException if term number “1” already exists for the policy. If you omit the term number, BillingCenter uses the number “1” for term number. If a policy period with the term number 1already exists then BillingCenter will throw an exception.

Workaround – Include a term number that is unique for the given policy number when sending a rewrite billing instruction.

Got exception when trying using the agency bill payment wizard to execute a promise with no dist items associated it (BC-4348)

Issue – Currently, if you click Execute on the Execute Agency Promise As Payment screen to apply a payment to a promise that has been pre-filled with zeroes, the following exception occurs:

EvaluationException: Cannot re-add AgencyCyclePayment since it has already been removed.

Workaround – Use the Payment Wizard to apply the payment instead.

AGBL statement Total Payment Applied is not correct after change 1 of 2 producer policies payment schedule (BC-2966)

Issue – The producer statement does not show the correct dollar amount after a change is made to one of the policy’s payment plans on a policy with two producers after the statement was billed and partially paid.

Workaround –

Guidewire is aware of this issue.

[After PaymentPlan Refactor, Payments are not distributed correctly across commissions \(BC-11892\)](#)

Issue – An issue occurs when performing a payment plan refactor on an agency bill policy that has been transferred. When redistributing agency bill payments, after payment plan change, BillingCenter calls the AgencyCycleDistPlugin plugin method `allocateForRedistribution`.

The base configuration of this plugin method uses the `gw.api.web.payment.ProRataRedistributionStrategy` class which does not reallocate commission fully in certain situations.

Workaround – The correct behavior is that, for each agency bill payment, the sum of commission amounts across all items targeting a specific charge should be equal before and after a payment plan change.

Contact Guidewire Customer Support for assistance in configuring the `allocateForRedistribution` method in the `AgencyCycleDistPlugin` plugin so that sum of commission amounts from each payment towards each charge matches the amounts contained in the allocation pool.

As a temporary alternative solution, if you encounter this problem, you can also manually redistribute the commission.

[Upgrade from BillingCenter 2.1.10 with an agency bill statement with a commission remainder invoice item may result in incorrect written off and net amounts for that agency bill cycle \(BC-10756\)](#)

Issue – If you are upgrading from BillingCenter 2.1.10 and you have an agency bill statement with a commission remainder invoice item, the upgraded written off and net amounts on the agency bill cycle can be inaccurate. The agency bill statement will correctly reflect the commission remainder. Only the agency bill cycle has the incorrect (not adjusted by commission remainder write-off) amounts for net and written off amounts.

Workaround – Guidewire is aware of this issue.

[When an invoice is carried forward, invoice items may be moved to an invoice on the wrong invoice stream \(BC-10505\)](#)

Issue – Note: This issue is not new to BillingCenter 7.0.5, but was newly discovered in this release. When an invoice is carried forward the invoice items may be moved to an invoice from the wrong stream, if there is more than one stream.

Workaround – To avoid this problem, update the `getCustomInvoiceItemPlacements` method in the `InvoiceAssembler` plugin as follows.

This code checks the invoice stream of the current invoice to see if it is the same as the invoice stream of the carry forward invoice. If the invoice streams are different, it searches for the next available invoice on the current stream for placement.

```
override function getCustomInvoiceItemPlacements(
    defaultPlacements : InvoiceItemPlacements, context : InvoiceAssemblerContext) : InvoiceItemPlacements
{
    if(context == "carriedforward")
    {
        for (invItem in defaultPlacements.InvoiceItems)
        {
            // if the stream of current is not the stream of the next
            if(invItem.Invoice.InvoiceStream >> defaultPlacements.invoiceFor(invitem).InvoiceStream)
            {
                // try to find the next invoice to place it on
                var nextInv = invitem.Invoice.InvoiceStream.Invoices.sortBy(\ i -> i.EventDate).
                    firstWhere(\ i -> i.EventDate > invitem.Invoice.EventDate)

                if(nextInv != null)
                {

```

```

        defaultPlacements.place(invItem).on(nextInv)
    }
}
}
return defaultPlacements
}

```

Studio/Platform Issues

Studio Rules do not use correct capitalization for name of root object (PL-10740)

Issue – Rule set root objects are expected to begin with an upper-case letter and issues warning if a rule set root object does not.

Workaround – You can safely ignore the warning message.

GX model generated XSD cannot be parsed by JAXB (PL-13598)

Issue – XSD generated by the GX model cannot be parsed by JAXB.

Workaround – Add JAXB annotation elements to the XSD to specify the necessary metadata, such as class names, to enable JAXB to generate the Java class files. Contact Guidewire Support for a sample XSD file that is annotated this way.

Problem with regen-java-api command and JAR files (PL-16351)

Issue – If you run the *BillingCenter/bin/gwpc regen-java-api* command, BillingCenter creates a *BillingCenter/java-api/lib* directory and puts JAR files for Java development in this directory. However, BillingCenter does not always perform this task correctly, resulting in some of the generated JAR files not being copied to this directory.

Workaround – For missing library files, copy or import into your Java development environment the missing library files from the following directory:

BillingCenter/admin/lib

Copy them into the following directory:

BillingCenter/java-api/lib

Renaming method or property throws ParseResultsException (PL-16633)

Issue – If you rename a property or a method or you change a method signature, and a workflow references that property or method in a Gosu field, BillingCenter throws *ParseResultsException*. This is the intended behavior.

Workaround – Restart the workflow engine. To do so:

1. Log into BillingCenter using an administrative account.
2. Access Internal Tools → Reload.
3. Click Reload Workflow Engine.

In some languages, web browsers render column headers of list views improperly (PL-18027)

Issue – In some languages, web browsers render some column headers of list views improperly if their column widths are specified too narrowly in their PCF definitions. For example, sometimes a numeric column is specified with a variable width of 1%. This narrow setting forces the browser to render the column too narrowly for the text of the translated column heading.

Workaround – Edit the PCF file that defines the column and clear the value from the width property. Without a specified value for the column width, browsers render the column widely enough to display the full text of the translated column heading.

[Database upgrade does not handle nullable to non-nullable columns with a default value for sub-types \(PL-23104\)](#)

Issue – For entity definitions, the automatic database upgrade converts nullable columns to non-nullable with a default value successfully. However, this column type conversion is not possible for columns in subtype definitions. BillingCenter implements non-nullable columns on subtype in the database as nullable because that column must have null values for rows that represents instances of other subtypes.

Workaround – Write a version trigger to populate the column with the default value for existing rows for the subtype. After you upgrade, BillingCenter enforces the column value to be non-nullable with the default value for new rows of the subtype.

[New transport plugin definitions do not show in the list of valid transport plugins \(PL-23317\)](#)

Issue – Newly implemented transport plugin definitions do not show in the list of valid transport plugins displayed by clicking the light-bulb icon next to the Transport Plugin field in the messaging destination editor.

Workaround – Restart Studio.

[Find in Resources fails for resources under Data Model Extensions or Web Resources \(PL-23320\)](#)

Issue – In Studio, the Find in Resources option does not work for resources that are in Data Model Extensions or Web Resources.

Workaround – To see the full resource path, hover over the resource name in the tab of the resource editor. Then, navigate the resource pane on the left to find the resource in the resource hierarchy.



chapter 24

Guidewire BillingCenter 7.0.4 Release Notes

Release 7.0.4

Release Notes Update: 22-July-2014

IMPORTANT These release notes replace the release notes that were included in the official product release. Please disregard the earlier version of the release notes.

Document Revision: 02-July-2015

Copyright © 2001-2015 Guidewire Software, Inc. All rights reserved.

Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire Reinsurance Management, Guidewire ContactManager, Guidewire Vendor Data Management, Guidewire Client Data Management, Guidewire Rating Management, Guidewire InsuranceSuite, Guidewire ContactCenter, Guidewire Studio, Guidewire Product Designer, Guidewire Live, Guidewire DataHub, Guidewire InfoCenter, Guidewire Standard Reporting, Guidewire ExampleCenter, Guidewire Account Manager Portal, Guidewire Claim Portal, Guidewire Policyholder Portal, Gosu, Deliver Insurance Your Way, and the Guidewire logo are trademarks, service marks, or registered trademarks of Guidewire Software, Inc. in the United States and/or other countries.

All other trademarks are the property of their respective owners.

This material is confidential and proprietary to Guidewire and subject to the confidentiality terms in the applicable license agreement and/or separate nondisclosure agreement.

Guidewire products are protected by one or more United States patents.

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release

- Support
- Issues and Major Changes
- Improvements and General Issues
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire BillingCenter.

Version Number

This release of Guidewire BillingCenter is 7.0.4

- For Guidewire Reporting, BillingCenter 7.0.4 supports IBM Cognos Business Intelligence 10.1 as the report server.
- If you are integrating with Guidewire PolicyCenter, this version of BillingCenter requires PolicyCenter 7.0.4 or later.
- If you are integrating with Guidewire ContactManager, this version of BillingCenter requires ContactManager 7.0.0 or later. Guidewire recommends that you use the latest release available, which is currently ContactManager 7.0.5.

Installing This Release

Refer to the following:

- The *BillingCenter Installation Guide* for general installation information
- Prior BillingCenter release notes for any versions that you have skipped

Support

For assistance, visit the Guidewire Resource Portal – <http://guidewire.custhelp.com>

Issues and Major Changes

This section contains issues and major changes that may affect your installation.

- Upgrade Support
- Base PCF File Changes
- Changes in This Release Provided in Upgrade Diff Report

Upgrade Support

Upgrading from BillingCenter 2.1.9 or newer versions is supported in this release.

The following information will be added to the *Upgrade Guide* in future releases:

Note: In cases where the definition of an Entity has been changed from being retireable to being non-retirable, BillingCenter 7.0.3 and later versions will not automatically remove data rows that had been retired. Instead, before upgrading to BillingCenter 7.0.3 or later versions, a new version check, `MigrateTableFromRetireableToEditableVersionCheck`, will run. This version check will log an error if any retired rows are found. If any such errors are logged, the retired rows must be examined and removed or restored before upgrade can continue.

Base PCF File Changes

All links below require the `readme_files` directory on your local disk.

BillingCenter release 7.0.3 to 7.0.4

- To view a report of the changes in the base PCF files in the `modules/bc` directory, [click here](#).
- To view a report of the changes in the base PCF files in the `modules/p1` directory, [click here](#).

Changes in This Release Provided in Upgrade Diff Report

Guidewire provides a report detailing certain differences between the current release and your prior release. This report describes changes in display keys, entities, typelists, and the Gosu API. To obtain your custom Upgrade Diff Report, visit the Guidewire Resource Portal.

Improvements and General Issues

BillingCenter Improvements and General Issues

The following are the improvements added to and the general issues corrected in this release:

ID	Description
Account	
BC-11122	<p>The method <code>Account.findAllPaymentMoneysDescending</code> returns a query that will return <code>DirectBillMoneyRcvd</code> objects. However, the <code>Query</code> interface does not support sorting. This means that the <code>DirectBillMoneyRcvd</code> objects will not be sorted in descending order by received date. You can add a sort order only after the <code>Query</code> has been transformed into an <code>IQueryResult</code> or <code>IQueryBeanResult</code> (via <code>select()</code> or <code>Processor</code>).</p> <p>Because of the return type, <code>findAllPaymentMoneysDescending</code> cannot be updated to have the behavior implied by the name of the method. Instead it has been deprecated, and a new method called <code>findAllPaymentMoneys</code> has been added. The <code>findAllPaymentMoneys</code> method does exactly the same thing as the now deprecated <code>findAllPaymentMoneysDescending</code>, but the name is more accurate.</p> <p>Before this change, the method <code>Account.redistributeAccountUnappliedFunds</code> used <code>findAllPaymentMoneysDescending</code> internally and, as a result, processed the <code>DirectBillMoneyRcvd</code> payments in a random order. This behavior has been changed. If funds tracking is enabled, <code>Account.redistributeAccountUnappliedFunds</code> will now process <code>DirectBillMoneyRcvd</code> payments in ascending order (oldest first). Otherwise <code>Account.redistributeAccountUnappliedFunds</code> will process <code>DirectBillMoneyRcvd</code> payments in descending order (newest first).</p> <p>If you are upgrading from BillingCenter 3.0.0 or earlier and funds tracking is disabled, the new behavior will match the original behavior. If funds tracking is enabled, processing the payments from oldest to newest makes more sense because funds tracking uses that order by default.</p>
Administration	
BC-10895	<p>Before this change, payment plan overrides were not properly exported to <code>admin.xml</code> and were lost when importing from that file.</p> <p>This issue has been addressed and payment plan overrides are properly included in the export.</p>
Agency Bill	

ID	Description
BC-10475	<p>Addressed an issue that occurred in BillingCenter 7.0.3 where commission was unearned.</p> <p>This happened in the following situation:</p> <ul style="list-style-type: none"> • Agency bill payment was reversed • Reversed payment included an item-level write-off • Commission earning criteria for the producer role was Invoice Fully Paid
BC-10765	<p>To improve the performance of the <code>AgencyPaymentSelectProducerPopup</code> page, rather than creating a <code>AgencyBillStatementView</code>, BillingCenter now uses a <code>StatementInvoice</code> entity.</p> <p>A new property named <code>AllInvoiceItemsExactlyPaid</code> was also added to the <code>StatementInvoice</code> entity. This property is true if gross and commissions are settled on every invoice item.</p> <p>This denormalized field removes the need to check for this during page load.</p>
BC-10774	<p>There was an issue where the select statement drop-down on the Agency Promise start page was not including write-offs in the Net Due calculation.</p> <p>This issue has been addressed.</p>
BC-10965	<p>There was an issue on the producer Agency Bill Open Items screen. If invoice items had been billed and written off, they still appeared on the Agency Bill Open Items screen.</p> <p>This screen has been updated to address this issue. Invoice items that have their remaining balance written off will no longer appear on the Agency Bill Open Items screen.</p>
BC-11059	<p>The Agency Payment Wizard was failing if there were already 2000 or more items on the payment and the Add Items button was used to find more items to include. This problem has been fixed by requiring you to filter by policy period when searching for more items to add, if there are already 2000 or more items included.</p>
BC-11252	<p>There was a issue with the calculation of earned commission on non-primary or non-active producers when the gross amount of an invoice item was only partially paid by an agency bill payment. Note that this problem occurred for partial agency bill payments only.</p> <p>This issue has been addressed.</p>
BC-11792, BC-11795	<p>Before this fix, there was an issue with primary commission transfers on billed agency bill invoice items. These commission transfers resulted in an assignment of the invoice item from one of the source producer's invoices to one of the target producer's invoices. Because the invoice item was billed, the assignment was done by reversing the original invoice item (which creates a new reversal invoice item on the original producer), and the creation of a new onset invoice item onto the target producer.</p> <p>The correct behavior is for the original invoice item and its reversal to continue to earn in the primary role on the source producer, but its item commission amounts should be moved to the onset item. The onset item then actively earns in the primary role on the target producer.</p> <p>Instead of the correct behavior, BillingCenter was incorrectly creating an item commission for the target producer on the original and reversal invoice items and marking it as the active primary earner. This is not allowed (agency bill invoice items must always be actively earning in the primary role on the producer whose statement invoice it is on), but it was not causing any errors because it was on a reversed and reversal invoice item, which were almost always filtered out of other BillingCenter operations.</p> <p>Because of changes in BillingCenter 7.0.4, these item commissions are no longer filtered out of BillingCenter operations, so it is now important that these erroneous item commissions be removed. BillingCenter will no longer create these erroneous item commissions.</p> <p>If you are upgrading from an earlier BillingCenter release, contact Guidewire Customer Support for assistance in finding and removing any erroneous item commissions. It is important to correct any such data before upgrading to BillingCenter 7.0.4 or later releases.</p>
Cancellation	
BC-9978	<p>Before this change, cancelling a policy period did not affect the policy period <code>ChargeProRataTx</code> entities. As a result, the policy period was prevented from closing until the <code>earnDate</code> of each <code>ChargeProRataTx</code> entity had passed.</p> <p>This issue has been addressed. Cancellation now collapses the outstanding <code>ChargeProRataTx</code> entities, setting each <code>earnDate</code> to the cancellation date.</p>
Commission	

ID	Description
BC-10168	<p>Adding a commission override after a commission write-off caused the commission reserve amount to be calculated incorrectly. The commission override calculation ignored the commission write-off, resulting in an incorrect Commission Reserve T-account balance.</p> <p>This problem has been fixed.</p>
BC-10347	<p>Addressed an issue where an error could occur during commission redistribution, if a positive and negative invoice item are added to a policy that earns with criterion On Binding and has had a commission transfer in the past. Now the commission is correctly distributed to the new invoice items, and no error occurs.</p>
BC-10669	<p>Previously when multiple small payments were made toward invoice items, it was possible that a penny could get stuck in reserve due to rounding.</p> <p>This issue has been addressed. Now, when an invoice item is fully paid BillingCenter checks for any remaining unearned commission, and ensures that all reserved commission becomes earned.</p>
BC-11344	<p>Addressed an issue where CommissionAdjustment invoice items were created when a commission override was done on a charge for a role other than primary. This issue has been fixed.</p> <p>CommissionAdjustment invoice items will only be created when a commission override is done on a charge for the primary role.</p>
BC-11553	<p>In previous versions of BillingCenter, calling <code>InvoiceItem.overrideCommissionRate()</code> would always round down the result of applying a new commission rate to an item, even if a different <code>RoundingMode</code> was specified in configuration files. This has been addressed by adding a new method that allows you to explicitly provide a rounding mode.</p>
BC-11602	<p>Calling the <code>InvoiceItem.remove()</code> method will now throw an <code>UnsupportedOperationException</code> if that item has been billed, has earned commission, or has had commission written off against it. This is a new safeguard to prevent data corruption.</p> <p><code>InvoiceItem</code> has a new method, <code>createAndPlaceOffset()</code>, which reverses that invoice item instead of removing it. Guidewire recommends that before calling <code>InvoiceItem.remove()</code> you first call <code>canRemove()</code>, and if the result is false, call <code>createAndPlaceOffset()</code> instead.</p>
Consistency Check	
BC-10597	BillingCenter consistency checks were improved to reduce false positives caused by switching roles on a contact.
BC-10598	BillingCenter consistency checks were improved to reduce false positives when a producer contact has multiple roles.
BC-10961	<p>The consistency checks in <code>ItemCommissionDenormFieldsDBCheckBuilder</code> were under-reporting errors because the SQL did not check the case where no roll-up transactions existed.</p> <p>This issue has been addressed and BillingCenter now verifies the denormalized fields in the case where no roll-up transactions exist.</p>
ContactManager Integration	
BC-11154, BC-11517	<p>Addressed an issue where sending vendor data that included tax filing status information from ContactManager to BillingCenter caused an error.</p> <p><code>ContactIntegrationXmlMapper.gs</code> has been updated to remove the <code>TaxFilingStatus</code> field of contact from the mapping of ContactManager data to BillingCenter data.</p> <p>If vendor data that includes tax filing status information is sent to BillingCenter from ContactManager, BillingCenter will now ignore the tax status portion of the data. Before this change, an error would occur.</p>
BC-11519	<p>Addressed an issue where you could not view and edit the files <code>bc-to-cm-type-mapping.xml</code> and <code>cm-to-bc-type-mapping.xml</code> in Guidewire Studio.</p> <p>You can now view and edit these files in Guidewire Studio.</p>
Delinquency	
BC-10803	Before this change, The Account Evaluation screen account delinquency summary field was only including policy-level delinquencies. The Account Evaluation screen now includes account-level delinquencies as well as policy-level delinquencies.
Disbursement	
BC-9400, BC-9490	Before this change, you could click Approve on rejected agency bill disbursements. This should not be allowed, and caused an error. Now the approve button is hidden for rejected disbursements.

ID	Description
BC-10894	Addressed an issue that caused an unnecessary copy of a payment with a null LoadCommandID to be created for some imported payments. This occurred when the Automatic Disbursements batch process attempted to distribute a payment that was imported from a staging table. The original payment was retired when the corresponding copy was created. This behavior did not affect correctness. After this fix, BillingCenter does not unnecessarily copy and retire payments.
Invoicing	
BC-10524	Addressed an issue where payment plans that based the installment date on ChargeDate failed to create the final installment if the final installment would have been on an invoice date. Payment plans using the ChargeEffectiveDate had the same problem for endorsements submitted with a BillingInstruction with a ModificationDate that was not at midnight. This issue has been addressed.
BC-10890	A new API has been added to the BillingAPI class to allow preview invoices for endorsements (policy changes). This new API takes information about a policy change and returns a preview of the new payment schedule. The new method, <code>previewInstallmentPlanInvoices(PolicyChangeInfo) : InvoiceItemPreview[]</code> , takes a PolicyChangeInfo object and returns an array of InvoiceItemPreview objects, each of which represents the total amount due for an invoice.
BC-11364	When default invoice items passed into the Payment Plan plugin function <code>createFullSetOfInvoiceItems</code> were discarded (i.e., not returned), BillingCenter would mistakenly attempt to examine those items, resulting in an exception. This has been addressed so that discarded invoice items are completely ignored.
BC-11567	When you change a policy payment plan and choose the Refactor All Invoice Items option, if BillingCenter was not able to reverse all of the invoice items an error occurred. Invoice items that cannot be reversed can be created as follows: <ul style="list-style-type: none"> • The policy is transferred to the current producer from another producer. • An invoice item was paid before the policy was transferred to the current producer from the original producer. The issue has been addressed as follows: BillingCenter creates new invoice items as of the date of the most recent irreversible invoice item. In a few cases, a positive down payment adjustment will result. The most common case is an irreversible down payment invoice item. Negative down payment adjustments will not be created. Instead, the difference will be distributed pro rata across the new invoice items. As a result, in the described scenario, the special case invoice items follow the down payment adjustment behavior of Planned Invoice Items Only rather than Refactor All Invoice Items .
BC-11686	Addressed an issue where BillingCenter allowed Gosu code to call <code>remove()</code> on an InvoiceItem, without adjusting the other invoice items associated with the charge before committing bundle. Normally BillingCenter verifies that the amounts of the invoice items on a charge added up to the amount of the charge, but BillingCenter was not doing this verification in this case. Now we correctly throw an exception if removing an InvoiceItem causes an inconsistency.
BC-11711	Addressed a problem where, if you attempted to change the payment schedule on a policy with reversed charges, BillingCenter would throw an exception and the payment schedule would not be successfully changed. Now if you change the payment schedule on a policy with a reversed policy charge, BillingCenter does not reschedule the reversed and reversal charges and the payment plan change occurs without any errors.
Messaging	
BC-11472	The Messaging.eix and Messaging.etx files contain indexes to speed up the messaging sub system when it searches for messages with a particular type of primary object, such as Contact, Account, Producer or PolicyPeriod. The existing Message.eix and Message.etx indexes have been improved, resulting in faster query plans and message processing. If you do not set the primary object on any messages, you can probably omit these indexes by deleting them from Message.etx. The one exception is the pair of indexes containing ContactID, which are used by the default contact integration and are therefore declared in Message.eix, which is not editable. You can still remove these indexes if necessary, by using the <code>remove-index</code> element in the Message.etx file.
Payments	

ID	Description
BC-10760	The Outgoing Producer Payments page has been updated to add a search button to the Payee input field. You now have the option of using the producer search page to select a producer for your Outgoing Producer Payments search.
BC-11178	<p>The Payment Request batch process has been updated to allow performance improvements, particularly for customers where this process has to handle a lot of work items and where the configuration added to this work queue (such as pre-update rules or messaging operations) is not too expensive.</p> <p>There is a new configuration parameter, <code>PaymentRequestsPerWorkItem</code>, which controls the number of payment request ids stored in a work item.</p> <p>The default value for <code>PaymentRequestsPerWorkItem</code> is 50.</p> <p>If you want the legacy behavior (BillingCenter 7.0.3 and earlier), set this parameter to 1.</p> <p>If you are taking advantage of this new parameter, consider that the optimal number of ids per work item depends on the cost of processing one work item, the total number of payment requests to be processed, and the number of worker threads used. Smaller numbers of payment requests per work item allow greater parallelism but can increase the costs of administering the queue (more queue items cause more locking). Larger numbers of payment requests reduce work queue administration costs, but reduce parallelism. Typically, if there are a large number of work items and each item uses a relatively small amount of processing time, storing multiple ids per work item will improve performance. The Payment Request work item cost is small in the default BillingCenter implementation. However, messaging costs may become much higher based on your configuration, so you may need to reduce the value of this parameter if you have significantly increased the cost of processing each work item.</p>
Performance	
BC-9844	<p>Prior to this change, you could search for accounts, policy periods, producers, invoices, trouble tickets, or payments based on just an address (such as 123 Main street). Searching by address only, without including additional search criteria, can be very slow.</p> <p>To address this issue, the <code>isReasonableConstrainedForSearch</code> method in <code>ContactCriteria.gs</code> has been updated to require additional restrictive search criteria. You now must enter more than an address to search for accounts, policy periods, producers, invoices, trouble tickets, or payments.</p>
BC-10793	<p>The <code>ReleaseTktHoldTypes</code> batch process now checks to see if it needs to update the policy or account hold denormalized fields as the result of a hold release. If any of the following consistency checks fail, please contact Guidewire Customer Support for assistance in repairing existing data:</p> <ul style="list-style-type: none"> • PolicyPeriod found with 'Held For Invoice Sending Denorm' true and no existing <code>InvoiceSending Trouble Ticket Hold</code> • PolicyPeriod found with 'Held For Invoice Sending Denorm' false and one or more existing <code>Invoice-Sending Trouble Ticket Holds</code> • PolicyPeriod found with the <code>ChargeHeld</code> property set to false and an existing held Charge • PolicyPeriod found with the <code>ChargeHeld</code> property set to true and no existing held Charge • Account found with <code>ChargeHeld</code> property not properly reflecting the state of held charges on the account
BC-10845	<p>The Agency Bill Payments page has been updated to improve performance, as follows:</p> <ul style="list-style-type: none"> • This page now uses a new view, <code>AgencyBillPaymentView</code>, that loads only the necessary information for the page. • The <code>Transaction Number</code> and <code>Distributed</code> columns have been removed from the Payments list view at the top of the Agency Bill Payments page.
BC-10862	<p>The Agency Bill Cycles page has been updated to improve performance, as follows:</p> <ul style="list-style-type: none"> • The <code>OpenOrPlanned</code> filter has been replaced by <code>Planned</code>. The Agency Bill Cycles page no longer shows open cycles. • Planned cycles now display all cycles for future statements, regardless of whether or not they are paid. Previously, if a planned statement was fully paid it was not included by the <code>OpenOrPlanned</code> filter. • The promise associated with the statement cycle is now cached on the Agency Bill Cycles page. This allows the page to load faster for cycles with promises.

ID	Description
BC-10916	<p>Updated the Agency Bill Statement page as follows. The Commission column now accurately displays the primary active commission amount for the associated invoice item. Previously, this page did not use the active status of the item commission and summed all associated primary item commissions instead of correctly using only the primary active item commission.</p> <p>Updated the Policy Activity list view (AgencyPaymentsItemsLV) in the Agency Payment wizard, as follows. The <insert icon here in release notes> (Fill in default values for a row) button now correctly calculates and populates the commission percentage column. Previously, this button did not use the active status of the invoice item's associated item commission and summed all associated primary item commissions instead of correctly using only the primary active item commission.</p>
BC-10957	<p>Addressed an issue with the Producer Statement page performance when a producer has a large number of item events.</p> <p>The list view (ItemEventLV) used to display item events on the Producer Statement page (ProducerStatement.pcf) has been updated. The list view now displays a list consisting of the Policy Number, Number of Earned Items, and the Commission Amount aggregated by policy number. A policy number search field was also added to the new list view. The policy number now links to a new popup called ProducerStatementPopup which contains the same level of detail previously included on the Producer Statement page, except that the content is now filtered by the policy number selected.</p>
BC-10998	<p>The Producer Policies detail and summary tab have been updated to improve performance.</p> <p>The Account evaluation on the producer policy detail and summary tab has been removed.</p>
BC-11055	<p>The performance of the policy closure batch process has been improved. The changes are as follows:</p> <ul style="list-style-type: none"> The policy closure batch process now filters out any policy periods with a remaining balance before it creates work items. Performance is improved because BillingCenter will now never create and process a work item for a policy period that has a remaining balance. Before this change, BillingCenter would create a work item and then, on processing that item, exit because the remaining balance prevented policy closure. However there is a <code>policyClosureConditionNotMet</code> method on the <code>IEventHandler</code> plugin. BillingCenter was calling this method to check whether policies with a remaining balance could be closed. Now the plugin will not be called in such a case because the work item is never created in the first place, in order to improve performance. It is important to note this change and update your configuration accordingly if you previously relied on this plugin being called to check whether policies with remaining balances could be closed. When a policy period is ready to be closed, BillingCenter no longer refreshes commission earnings on all policy commissions for that policy period. Instead, BillingCenter now only refreshes commission earnings for policy commissions with payable criteria of Effective Date, Expiration Date, or Custom. BillingCenter no longer refreshes commission earnings for policy commissions with any other payable criteria, because in all other cases commission is earned immediately when the criteria event occurs. Commission earnings are always up to date (and refresh is unnecessary) for all earning criteria with the exception of: Effective Date, Expiration Date, and Custom.
BC-11061	<p>BillingCenter 7.0.4 includes a new logger category, <code>Application.InvoiceWorkQueue</code>. If you want the same level of logging for invoice batch processes as you had in BillingCenter 7.0.3 then set this category to use DEBUG logging. For example, in <code>logging.properties</code>:</p> <pre>log4j.category.Application.InvoiceWorkQueue=DEBUG</pre>
BC-11477	<p>Added an internal performance optimization in the way BillingCenter writes some changes to the database. BillingCenter now uses batch updates for updating denormalized amount properties, and also bypasses the normal optimistic locking mechanism when it is safe to do so.</p> <p>This is a significant performance improvement but does mean that the entities for which the only change is the update of a denormalized amount field will no longer be on the list of modified entities in the bundle and will not fire events</p>

ID	Description
BC-11481	<p>Bundle validation has been disabled for BillingCenter specific work queues. Bundle validation is a standard Guidewire platform feature which refreshes each bundle of entity changes and runs validation rules on every commit. This bundle refresh re-fetches every unchanged entity that is in the bundle. BillingCenter does not provide any validation rules in the default configuration.</p> <p>Validation rules are only supported on certain platform provided entities (Activity, Contact, Group, User, and Region).</p> <p>Performing refresh and validate is normally unnecessary for BillingCenter specific batch processes, as they are very unlikely to alter any of the entities that support validation. It is also very unlikely that any bundle committed by a work queue needs a refresh, as it probably has an up to date copy of the entities. Even though BillingCenter does not have any validation rules, there is still some expense to validating every bundle commit – in particular, the refresh of all the entities can result in extra database queries. Therefore, we have disabled bundle validation for BillingCenter specific work queues. If you do have validation rules, and you require them to run for the BillingCenter specific work queues, then you can set the new configuration parameter, <code>EnableWorkQueueValidation</code>, to true. Guidewire does not recommend this, particularly if your nightly work queues take a long time.</p> <p>Note that this change does not affect the standard work queues provided as part of the Guidewire platform, such as Activity Escalation, Group/User Exception and Contact Auto Sync. These work queues will continue to use validation, as in previous releases of BillingCenter.</p>
BC-11504	<p>The default value of the <code>KeyGeneratorRangeSize</code> configuration parameter has been increased from 100 to 1000.</p> <p>This parameter is the block size used by the key allocator, used to create the ids for newly created entities. The allocator allocates a block of keys (using a DB lock, so nobody else in the cluster will use that same block) and then supplies keys from that block. The larger the block size, the less often the allocator has to go back to the database to get a new block.</p> <p>Possible impacts of this change are as follows.</p> <ul style="list-style-type: none"> It has always been possible for ids to be allocated out of order across a cluster. For example, one server might give a new entity the id 150 and then, later, another server might give a new entity of the same type the id 50 (this would happen if the first server was allocated from a block containing the ids 101-200 while the second was using the block 1-100). This effect will be more obvious with the new, larger, block size as there will be even larger differences between new ids allocated on different servers. Some number of ids are unused when a server is shut down, and these ids will never get used -- the unused ids in the current allocation block of the server become unavailable on shutdown. Increasing the block size increases the number of potential ids that become permanently unavailable for future use. However, the id space is much larger than the fraction of ids that will eventually become unavailable – less than one thousandth of one percent.
BC-11543	<p>BillingCenter has been updated to improve performance by eliminating certain concurrent data change exceptions that occurred because of updates to denormalized fields. This improvement allows BillingCenter to remove the Producer column from the Invoice Billed and Invoice Due batch processes. BillingCenter no longer restricts batch processes so that only one thread can change a Producer at any given time.</p>
BC-11548	<p>The Policy Closure batch process has been updated as follows to improve performance. This batch process now filters out all cancellations except flat cancellations before creating work items to be processed. Flat cancellation is the only type of cancellation which can close a policy without reaching the policy period expiration date</p>
BC-11569	<p>This change improves the performance of the Funds Allotment work queue by adding a denormalized bit column called <code>FullyAllotted</code> to the <code>bc_fundstracker</code> table. This column is set to 1 if, and only if, the <code>AmountAllottedDenorm</code> and <code>TotalAmount</code> columns are equal (both of these columns are non-nullable, so there are no complications with null values). The denormalized bit column is also incorporated into an index, which is used to speed up the queries used by the Funds Allotment work queue.</p>
BC-11620	<p>The <code>InvoiceItem</code> entity has two new fields:</p> <ul style="list-style-type: none"> <code>hasChargeBilledTransaction</code> – True if invoice item has a non-reversed <code>ChargeBilled</code> transaction <code>hasChargeDueTransaction</code> – True if invoice item has a non-reversed <code>ChargeDue</code> transaction <p>These fields were added to improve performance.</p>

ID	Description
BC-11570	<p>The Funds Allotment batch process has been updated to allow performance improvements, particularly for customers where this process has to handle a lot of work items and where the IFundsTracking.allotFunds plugin call performs well.</p> <p>There is a new configuration parameter, FundsAllotmentAccountsPerWorkItem, which controls the number of account ids stored per work item for the Funds Allotment work queue.</p> <p>The default value for FundsAllotmentAccountsPerWorkItem is 10.</p> <p>If you want the legacy behavior (BillingCenter 7.0.3 and earlier), set this parameter to 1.</p> <p>If you are taking advantage of this new parameter, consider that optimal number of ids per work item depends on the cost of the IFundsTracking.allotFunds plugin method, the total number of accounts to be processed and the number of worker threads used. Smaller numbers of accounts per work item allow greater parallelism but can increase the costs of administering the queue (more queue items cause more locking). Larger numbers of accounts reduce work queue administration costs, but reduce parallelism.</p>
Platform	
BC-11496 (PL-23427)	<p>Addressed an issue in which casting errors were thrown when Gosu code called a Java method that expected a specific entity, such as Document, as a parameter. The system would throw an error similar to the following:</p> <pre>gw.lang.parser.EvaluationException: com.guidewire.commons.metadata.proxy._generated.impl.Document cannot be cast to com.guidewire.pc.external.entity.Document</pre> <p>This error was caused by the way in which Guidewire Java code was making a reflective method call to Gosu and then handling external entity types. Those reflective method calls no longer require casting of the external entity types.</p> <p>The signature of the IBatchCompletedNotification.complete() method has been changed as follows:</p> <pre>new: public void completed(ProcessHistory batch, int workItemsFailed); old: public void completed(BatchProcessType batch);</pre> <p>The new signature allows passing the number of failed work items.</p> <p>Note that before this change BillingCenter did not call complete() if there were failed work items in the queue. Now BillingCenter calls complete() if there are no work items other than the failed work items in the queue.</p>

ID	Description
BC-11679 (PL-23686)	<p>Generating database upgrade statistics during upgrade is now optional. Database upgrade statistics are not generated for Microsoft SQL Server or the H2 development database, so this change does not affect those databases.</p> <p>You set this new option in the config.xml file. In the <database> block, there is an <upgrade> block that contains configuration information for the overall database upgrade. There is a new attribute in the <upgrade> block, updatestatistics with default value true. If you set this attribute to false, statistics will not be updated for that database during upgrade.</p> <p>If you do not update statistics during upgrade, you are then responsible for updating statistics by using the command-line batch process. Most of the time, the incremental statistics update is sufficient. For example:</p> <pre data-bbox="479 536 1263 561">maintenance_tools -password password -startprocess incrementaldbstats</pre> <p>If statistics are not updated during the upgrade, you see a warning that recommends that you run the database statistics batch process in incremental mode. Additionally, the UpgradeInfo page shows that statistics were not updated as part of the upgrade. This page also reports the runs of the statistics batch process, and incremental runs are shown.</p> <p>The UpgradeInfo page does not identify the following case: You ran another upgrade with updatestatistics=true since running a previous update with Generating database upgrade statistics during upgrade is now optional. Database upgrade statistics are not generated for Microsoft SQL Server or the H2 development database, so this change does not affect those databases.</p> <p>You set this new option in the config.xml file. In the <database> block, there is an <upgrade> block that contains configuration information for the overall database upgrade. There is a new attribute in the <upgrade> block, updatestatistics with default value true. If you set this attribute to false, statistics will not be updated for that database during upgrade.</p> <p>If you do not update statistics during upgrade, you are then responsible for updating statistics by using the command-line batch process. Most of the time, the incremental statistics update is sufficient. For example:</p> <pre data-bbox="479 1015 1263 1041">maintenance_tools -password password -startprocess incrementaldbstats</pre> <p>If statistics are not updated during the upgrade, you see a warning that recommends that you run the database statistics batch process in incremental mode. Additionally, the UpgradeInfo page shows that statistics were not updated as part of the upgrade. This page also reports the runs of the statistics batch process, and incremental runs are shown.</p> <p>The UpgradeInfo page does not identify the following case: You ran another upgrade with updatestatistics=true since running a previous update with updatestatistics=false, but you did not update statistics first.</p> <p>When you click the Download button on the UpgradeInfo page, you get a more detailed UpgradeInfo report.</p> <p>The UpgradeInfo report shows the value of the updatestatistics attribute at the time of upgrade. Additionally, the UpgradeInfo report shows the update statistics SQL statements that were skipped as part of the upgrade. Normally, you do not need to consult this list because running the database statistics batch process in full mode will cover these statements. Note that the incremental mode might not capture all cases that would have been run during an upgrade with updatestatistics=true, but you did not update statistics first.</p> <p>When you click the Download button on the UpgradeInfo page, you get a more detailed UpgradeInfo report.</p> <p>The UpgradeInfo report shows the value of the updatestatistics attribute at the time of upgrade. Additionally, the UpgradeInfo report shows the update statistics SQL statements that were skipped as part of the upgrade. Normally, you do not need to consult this list because running the database statistics batch process in full mode will cover these statements. Note that the incremental mode might not capture all cases that would have been run during an upgrade with updatestatistics=true.</p>
BC-11680 (PL-23785)	Altered the algorithm used to calculate column statistics for database upgrade.

ID	Description
BC-11681 (PL-24640)	<p>Resolved an issue with the Gosu type system that led to an infinite loop and a stack overflow exception when constructing an entity type. The issue occurs in earlier releases if an entity uses <implementsInterface> and that interface has a method that returns an array of the same entity.</p> <p>Note that one consequence of this change is that the Java representation in Gosu byte-code are now the same for any two customer entities with the same base type, where base type is versionable, retireable, etc. This means that on a Gosu class or interface, any two methods whose signatures differ only by customer entity types will need to be renamed, as they resolve to the same byte-code signature. For example, given customer entities A and B that are both retireable, the following class will not compile:</p> <pre>interface Foo { function bar(a: A) function bar(b: B) }</pre> <p>The reason is that the Java byte-code representing this class will result in two methods with the same Java signature:</p> <pre>// Java byte-code representation of the Gosu interface "Foo" public interface _Foo { void bar(com.guidewire.commons.entity.EditableRetireable a); void bar(com.guidewire.commons.entity.EditableRetireable b); // Invalid: a method already exists with this signature }</pre> <p>The remedy in this case is to rename at least one of the methods. You may prefer to rename both for consistency:</p> <pre>interface Foo { function barA(a: A) function barB(b: B) }</pre> <p>Note that the following examples of overloading (two methods with the same name but different parameter types) are still OK, because there are other differences to make the byte-code signatures distinct:</p> <pre>interface Foo2 { function bar(a: A, n: int) function bar(b: B, s: String) // OK, since the methods also differ by the types of their second parameters } interface Foo3 { function bar(a: A, n: int) function bar(b: B) // OK, since the methods have different numbers of parameters }</pre>
Policy	
BC-11139	
BC-11162	<p>In previous versions of BillingCenter the default implementation of IPolicyPeriod.applyFullPayDiscount logged several lines at INFO level. Now these lines are logged at DEBUG level. If you are using the default implementation (which is not recommended, it is only provided as sample code) and you also want the old amount of logging, you can add a line like the following to your logging.properties file: log4j.category.Application.FullPayDiscountWorkQueue=DEBUG</p> <p>The Full Pay Discount batch process has been updated to allow performance improvements, particularly for customers where this process has to handle a lot of work items and where the IPolicyPeriod.applyFullPayDiscount plugin call performs well.</p> <p>There is a new configuration parameter, FullPayDiscountPolicyPeriodsPerWorkItem, which controls the number of policy period ids stored in a work item.</p> <p>The default value for FullPayDiscountPolicyPeriodsPerWorkItem is 10.</p> <p>If you want the legacy behavior (BillingCenter 7.0.3 and earlier), set this parameter to 1.</p> <p>If you are taking advantage of this new parameter, consider that the optimal number of ids per work item depends on the cost of the IPolicyPeriod.applyFullPayDiscount plugin method, the total number of policy periods to be processed and the number of worker threads used. Smaller numbers of policy periods per work item allow greater parallelism but can increase the costs of administering the queue (more queue items, more locking). Larger numbers of policy periods reduce work queue administration costs, but reduce parallelism.</p>
Policy Closure	

ID	Description
BC-11241	An exception could be thrown when closing a policy for which a charge negative write-off was reversed. This will no longer occur.
Producer	
BC-11564	The Producer Payment batch process has been updated to improve performance. This batch process now uses a bulk update operation to update denormalized fields on ItemCommissions, InvoiceItems and Producer entities. These entities will no longer fire rules or generate messages when the only fields changed are the denormalized fields.
Reinstatement	
BC-11549	Previously, when a policy was flat cancelled, and then reinstated, the policy closure status was not changed from CLOSED to OPEN. Now, after a flat-cancelled policy is reinstated, the policy closure status is set to OPEN or OPEN_LOCKED as appropriate.
Rewrite Process	
BC-11554	Before this fix, when a policy was flat cancelled and then rewritten, the policy closure status was not changed from CLOSED to OPEN. Now, after a flat cancellation followed by a rewrite, the policy closure status is set to OPEN or OPEN_LOCKED as appropriate.
Staging Tables	
BC-11667	Negative write-offs can now be loaded from staging tables. Note: The NegativeWriteoff.ExecutionDate field must be populated in the staging table but there is no integrity check to warn if the ExecutionDate is null. A null ExecutionDate will cause an exception when trying to load from staging tables.
Transactions	
BC-10407	Reversing a charge correctly created reversal invoice items and ChargePaidFromAccount transactions for the reversal invoice items. The ChargePaidFromAccount transactions also correctly had ChargePaidFromAccount.InvoiceItem.Charge set to the reversal charge, but they incorrectly had ChargePaidFromAccount.Charge set to the original charge. Now reversing a charge also correctly sets ChargePaidFromAccount.Charge to the reversal charge.
BC-11194	Addressed an issue which could create duplicate T-Accounts in a rare race condition in BillingCenter 7.0.3. This change should be invisible to you, but there is a small possibility that you already have duplicate data. You can check for this possible duplicate data using the following query: select dupcount, count(dupcount) from select taccountcontainerid, taccountpatternid, count(*) dupcount from bc_taccount group by taccountcontainerid, taccountpatternid having count(*) > 1 sub group by dupcount; If this query returns any results, contact Guidewire Customer Support for assistance.
Upgrade	
BC-11184	Addressed the following issue that could occur during upgrade from BillingCenter 2.x to 7.x. Direct bill payments for account level charges can be modified (reversed) in BillingCenter 2.x without losing their distribution. These payments are still attached to the original charge paid transactions through the charge paid sources. However, during upgrade to BillingCenter 7.x the connection between these payments and the original distribution was being lost. The connection between these payments and the original distribution is now correctly retained after upgrade.
BC-11001	This change addresses a problem encountered during upgrade from BillingCenter 2 with a certain type of unsupported data. Specifically, the data that caused upgrade to fail consisted of direct bill charges having invoice items on statement invoices. In this case, the Upgrade Commission batch process was overwriting the commission amounts on these agency items. In order to allow upgrade from BillingCenter 2 to succeed, the Upgrade Commission batch process has been updated to ignore any agency bill items on direct bill charges. This change has no effect if you do not have this unusual data problem.

ID	Description
Write-off	
BC-11272	<p>BillingCenter does not write off commission for secondary or referrer producers when you write off commission for the primary producer. The <code>WriteOffFactory</code> class has a new method, <code>createInvoiceItemCommissionWriteOff</code>, that can be used to create write-offs which will trigger the appropriate approval process. Use this new method if you want to configure commission write-offs for the primary producer to automatically write off secondary and referrer commission.</p> <p>The following code illustrates the correct way to create and execute an <code>ItemCommissionWriteOff</code>:</p> <pre>gw.transaction.Transaction.runWithNewBundle(\ bundle -> { var writeOff = new WriteOffFactory(bundle).createInvoiceItemCommissionWriteOff(itemCommission, null) writeOff.doWriteOffWithApproval() })</pre>

Platform Improvements and General Issues

ID	Description
Build Infrastructure	
PL-15509	<p>In previous versions, Guidewire supported running build scripts only through the <code>gwbc</code> command on Windows. With this version you, can run the EAR and WAR build scripts on Unix by invoking Ant directly, with the following command:</p> <pre>ant -f BillingCenter/modules/ant/build.xml buildScript</pre> <p>Substitute <code>buildScript</code> with one of the following values:</p> <ul style="list-style-type: none"> • <code>build-jboss-war</code> – Builds a generic WAR file for use with JBoss • <code>build-tomcat-war</code> – Builds a generic WAR file for use with Tomcat • <code>build-weblogic-ear</code> – Builds an EAR file for use with WebLogic • <code>build-websphere-ear</code> – Builds an EAR file for use with WebSphere <p>Any other build scripts that you can run through the <code>gwbc</code> command must be run on Windows.</p>
Cognos Integration	
PL-22539	With this release, the Cognos plugin no longer removes domain names from the URL parameter to the <code>setCognosCookiesUrl</code> plugin method.
Command line tools	
PL-20378	With this release, the <code>verify-types</code> command-line tool does not stop verification if a failure, such as a programming error, occurs while verifying a specific type. The tool reports all errors, warnings and failures.
Database support	
PL-21842	With this release, the <code>system_tools</code> command has additional options for submitting batch jobs that report database performance.
PL-23523	For Oracle, you can configure new LOB columns to use SecureFile LOBs or compressed SecureFile LOBs instead of the default BasicFile LOBs. You can configure the LOB type for the entire BillingCenter database or for specific tables only. For more information, see “Configuring Oracle LOB Types” in the <i>Installation Guide</i> .
Database upgrade	
PL-22478	In earlier releases on SQL Server, queries failed if they contained a large number of CASE conditions. Now, BillingCenter breaks up queries with more than 125 CASE conditions into nested CASE clauses or into separate queries.
PL-22686	With this release, upgrade version checks now include validation SQL in the downloads that you obtain from the Upgrade Info page.
PL-22934	This release fixes an issue that caused a SQL failure during database upgrade.

ID	Description
PL-23044	With this release, the database upgrade now supports table and index partitioning. Specify which tables and indexes you want to have partitioned, and the upgrader creates any new table and new indexes appropriately. Existing tables and indexes remain unchanged, regardless of the discrepancy between the metadata configuration file and the database schema.
PL-23104	With this release, the automated database upgrade supports changing a column on an entity definition from nullable to non-nullable.
PL-23504	With this release, a database upgrade in a development environment records checkpoints of upgrade triggers that complete successfully. You can restart a failed database upgrade, and it resumes with the upgrade trigger that failed. The restart feature helps you test your upgrade with realistically large data sets. You avoid time spent to restore the database and time spent to run upgrade triggers that work successfully. To restart a test database upgrade from a checkpoint reached in an earlier upgrade, you need to roll back manually any database changes that occurred during the upgrade trigger that failed. In addition, assure you resolved the problem that caused the trigger to fail before restarting. A test run of your upgrade is successful only when it runs from start to finish without a restart. Never use the restart feature of database upgrade in a production environment.
Document Management	
PL-13904	A new class, DocTreeNode, simplifies the creation of a tree that is based on a customer defined path.
PL-22562	With this release, you can pass custom variables from document descriptor files to document templates and merge scripts. Custom variables let you add information to produced documents, including to their headers and footers, and they let you add conditional processing logic. First, declare the variable in the XSD for document template descriptors. Second, set the variable to a value in the metadata of the appropriate document descriptor file. Finally, use the variable, with an underscore (_) as a prefix, in the appropriate document template or merge script. For example, to add conditional processing to the production of Word documents: 1. Declare the variable in the XSD for document template descriptors – In Studio, open the file document-template.xsd. Locate this XSD by pressing CTRL+N and entering document-template. Add the variable as an XSD attribute, as the following sample code shows. <pre><xsd:attribute name="mail-merge-type" type="xsd:string" use="optional"/></pre> Then, save the file. 2. Set the variable to a value in the document template descriptor – In Studio, navigate to resources → Other Resources → doctemplates and open the file SampleLetter.gosu.rtf.descriptor. Add the variable as attribute on the <DocumentTemplateDescriptor> element, as the following sample code shows. <pre>mail-merge-type="catalog"</pre> Then, save the file. 3. Use the variable in the Mail Merge and Close merge script – In Studio, navigate to resources → Other Resources → Web Resources → resources → documentassistant. Open the file MailMergeAndClose.js. Modify the file to conditionalize the following Javascript statement, as the following sample code shows. <pre>if (_mail-merge-type == "catalog") { objWord.MailMerge.MainDocumentType = /* wdCatalog */ 3; }</pre>
PL-22645	With this release, users should be able to edit documents whenever IDocumentContentSource uses mode=url.
PL-22737	In previous versions, the BFO library was broken for some PDF templates. This version includes an updated BFO library that corrects the problem.
Entities/Metadata	
PL-22866	Calling an API method that attempts to modify a read-only bundle now displays an immediate error.
Management plugin	
PL-16543	With this release, the counters NumActiveDBConnections and NumIdleDBConnectionsmanagement are integers instead of strings, so management tools that connect via JMX can plot them now on graphs.
Persistence	

ID	Description
PL-23941	For workflow-related distributed work queues, the logging level of the message "WDW processing workitem: ..." changed from INFO to DEBUG to reduce noise in the log files.
Web - Other	
PL-18559	Added alt-text, or alternative tooltip text, for the following: <ul style="list-style-type: none">• BooleanRadioInput• PicketLink• DocumentationCell
Web – UI/Runtime	
PL-23848	PCF input elements now render currency amounts so the scale and appscale parameters are honored. For example, if appscale == 0 and the amount is the integer 1, the input element renders the amount as "1". If appscale == 2 and the amount is the integer 1, the input element renders the amount as "1.00".
Web services – WS-I	
PL-23514	This release introduces a new method, preExecute, for use in the invocation handlers that you write for WS-I web services. In earlier releases, if you wrote an invocation handler, you unavoidably bypassed some important WS-I features. For example, the application: 1) did not enable profiling for method calls; 2) did not check run level annotations, even at the class level; 3) did not check web service permission annotations, even at the class level; 4) did not check for duplicate external transaction IDs if those SOAP headers are present. Now, you can call the preExecute method in your invocation handler to assure the application takes the preceding actions, as needed. See the <i>Integration Guide</i> for more information about invocation handlers and how to use the preExecute method.
PL-23721	To detect duplicate operations from external systems, add the new annotation @WsCheckDuplicateExternalTransaction to your WS-I web service implementation class. See the <i>Integration Guide</i> for details for using the annotation and the SOAP header for the transaction ID.

Known Issues and Limitations

This section describes known issues with this release of Guidewire BillingCenter.

- BillingCenter Known Issues
- Studio/Platform Issues

BillingCenter Known Issues

Delinquency: Delinquency can be started on an account, even though there is no amount due, when negative charges are present ([BC-3746](#))

Issue – Delinquency can be started on an account when no amount is due if a negative charge on a policy is present.

Workaround – Invoke the doDueTransaction() method on the negative invoice item. This will execute a ChargeDue transaction for that invoice item which will prevent the problem.

Note that you can execute invoiceItem.doDueTransaction() method on a negative invoice item even if it is on an invoice that is in billed state not yet due. Later, when the invoice is made due, BillingCenter will just skip the negative invoice (because invoiceItem.doDueTransaction() already forced that invoice item to be due).

PolicyCenter Integration: BC Renewal thread fails with exception on issue after removal of Alt Billing Account from Renewal in PC. ([BC-9540](#))

Issue – If you are integrated with PolicyCenter, you will encounter an error if you renew a policy that had an alternate billing account and you remove the alternate billing account in PolicyCenter prior to committing renewal.

Workaround – Guidewire is aware of this issue.

Contact Manager: BC does not show updated Primary Address after CM contact Merge (BC-9315)

Issue – If you are using BillingCenter with Contact Manager, there is a problem with merging contacts with different addresses. The primary address of the merged contact is not shown in BillingCenter.

Workaround – Guidewire is aware of this issue.

Commission: Commission override after a commission write-off causes write-off to be reserved (BC-9536)

Issue – Several methods of the BillingAPI WS-I webservice, such as `issuePolicyPeriod()` and `cancelPolicyperiod()`, indirectly call `BillingInstructionUtil.executeAndCommit()` which commits the current bundle.

These methods also directly call `tryCatch()`, which commits a new bundle after executing the entire block of code inside of `tryCatch()`. As a result, the code tries to commit the same bundle twice.

Here are two ways you can encounter this problem:

1. If you try to create an activity within the block of code passed to `tryCatch()` after the call to execute the billing instruction, BillingCenter will throw a `DBA1readyExecutedException`.

For example:

```
function issuePolicyPeriod(issuePolicyInfo : IssuePolicyInfo, transactionId : String) : String{
    require(issuePolicyInfo, "issuePolicyInfo")
    var publicID = tryCatch( \ bundle -> {
        var pid= issuePolicyInfo.executeIssuanceBI()
        // code *here* to create a new activity
        return pid
    }, transactionId)
    return publicID
}
```

2. If you used `CreateTroubleTicketHelper` to create a trouble ticket in a billing instruction preupdate rule, you would again get a `DBA1readyExecutedException`. The following is an example of a billing instruction preupdate rule that would trigger the error:

```
// create trouble ticket with a delq hold until second invoice
var troubleTicketHelper = new CreateTroubleTicketHelper()
var ticket = troubleTicketHelper.createTroubleTicket((billingInstruction as
    PlcyBillingInstruction).PolicyPeriod)
ticket.Title = "Delinquency hold on issuance"
ticket.DetailedDescription = ticket.Title
ticket.Priority = "normal"
ticket.TicketType = "automatic"
ticket.Hold.setAppliedToHoldType(HoldType.TC_DELINQUENCY, true)
ticket.Hold.setReleaseDate(HoldType.TC_DELINQUENCY, gw.util.GWBaseDateEnhancement.CurrentDate)
```

Workaround –

1. For this case, in the block of code passed to `tryCatch()`, do not create or update entities after the call to execute the billing instruction. For example:

```
function issuePolicyPeriod(issuePolicyInfo : IssuePolicyInfo, transactionId : String) : String{
    require(issuePolicyInfo, "issuePolicyInfo")
    var publicID = tryCatch( \ bundle -> {
        // code *here* to create a new activity
        var pid= issuePolicyInfo.executeIssuanceBI()
        return pid
    }, transactionId)
    return publicID
}
```

2. For this case, instead of using `CreateTroubleTicketHelper` to create the trouble ticket, create related entities directly as in the following example in a PolicyPeriod enhancement:

```
var ticket = new TroubleTicket(this)
var ttJoinEntity = new TroubleTicketJoinEntity(ticket)
ticket.Title = ticketTitle
ticket.DetailedDescription = "Delinquency hold placed on Account with a " + ticket.Title
```

```

ticket.Priority = "normal"
ticket.TicketType = ticketType
ticket.Hold.setAppliedToHoldType( HoldType.TC_DELINQUENCY, true )
ticket.Hold.setReleaseDate(HoldType.TC_DELINQUENCY, releaseDate)
ticket.addToTroubleTicketJoinEntities(ttJoinEntity)
this.addToTroubleTicketJoinEntities(ttJoinEntity)
this.Account.addToTroubleTicketJoinEntities(ttJoinEntity)
ticket.Hold.checkForHoldAdditions()

```

[PolicyCenter Integration: BC Renewal thread fails with exception on issue after removal of Alt Billing Account from Renewal in PC \(BC-9540\)](#)

Issue – If you are integrated with PolicyCenter, you will encounter an error if you renew a policy that had an alternate billing account and you remove the alternate billing account in PolicyCenter prior to committing renewal.

Workaround – Guidewire is aware of this issue.

[PolicyCenter Integration On Renewal Offer, Suspense Payment Should Be Applied At Policy Period \(BC-9691\)](#)

Issue – When integrated with PolicyCenter, a suspense payment that is created with an offer number from a PolicyCenter renewal is getting applied to the owner account of the policy period. The suspense payment should be applied to the renewal policy period instead of the owner account.

Workaround – Guidewire is aware of this issue.

[Payments: Reversing Producer Payments is throwing NullPointerException when the payment is related to a retired invoice item \(BC-9749\)](#)

Issue – BillingCenter does not allow invoice items that have earned commission to be deleted. BillingCenter already prevents such items from being removed through the user interface. If an invoice item that has earned commission is deleted programmatically, errors will occur if the related payment item is reversed.

Workaround – Instead of deleting invoice items that have earned commission, set the invoice item amount to zero.

[Disbursement: Changing the Disbursement reason from Automatic causes it to be sent out despite a hold \(BC-9761\)](#)

Issue – The Disbursements batch process sends an disbursement if the disbursement reason is changed from 'Automatic' to something else, even if there is an active Trouble Ticket with a hold of type 'Disbursements' on the owning account. The BillingCenter user interface does not allow editing of automatic disbursements, but they can be edited through configuration. BillingCenter fails to account for this and the disbursement batch process treats a reason other than automatic to mean that it is a manual disbursement and so processes it, ignoring any holds.

Workaround – Use a custom extension to store more specific reasons for an automatic disbursement.

[Payment Distribution: BC should not distribute payment amount that have been disbursed \(BC-9783\)](#)

Issue – Reversing a DirectBillPayment is causing T-account Unapplied to become negative. This can happen when another direct bill payment on the same account gets modified.

Workaround – Create a recapture charge when unapplied goes below zero.

Changing Billing Method From AB to DB for earning on First Payment Received For Billed Statement Is Earning Commission (BC-10476)

Issue – Changing the billing method on a policy period from agency bill to direct bill with a producer that earns commission on First Payment Received will instead earn the commission when invoiced. Commission should not be earned until a payment is received.

Workaround – As part of transferring a policy period from agency bill to direct bill, add code to the policy period preupdate rule to reverse the earning transaction if the total amount paid on the policy period is zero. This will make the earning go back to zero for the policy period. The commission payable batch process will not identify these charge commissions (since their earning criteria will not match the selection criteria). When a real payment is posted to the policy, commission earning will be triggered by that event and be earned correctly.

PolicyNumberLongDenorm not updated in upgrade (BC-10742)

Issue – Prior to 3.0, BillingCenter used 0-based term numbers for policies. Starting with 3.0, BillingCenter uses 1-based term numbers for policies to be consistent with PolicyCenter term numbering. If you are not upgrading from a version prior to 3.0 this issue does not affect you.

Because the `PolicyNumberLong` field (and the corresponding `PolicyNumberLongDenorm` field) can be a legal reference, BillingCenter does not change the value during upgrade.

The upgrade does renumber the `TermNumber` from being 0-based to being 1-based, since the `TermNumber` is not normally used as a legal reference. This behavior is to avoid a gap between term numbers. Otherwise, the terms would have a gap. For example, instead of `TermNumber` 0, 1, <upgrade> 3, 4, the upgrade renames the first two to make it 1, 2, 3, 4.

However, because the upgrade does not adjust `PolicyNumberLong` and `PolicyNumberLongDenorm`, there will be a gap in the `PolicyNumberLong` (for example 123456-0, 123456-1, <upgrade> 123456-3, 123456-4, skipping any 123456-2 `PolicyNumberLong`).

Workaround – If you do not need to preserve your policy numbers after upgrade and you want your `PolicyNumberLong` and `PolicyNumberLongDenorm` values to match the new post-upgrade `TermNumber`, contact Guidewire Customer Support for information on how to update these fields.

IMPORTANT: The `PolicyNumberLong` and `PolicyNumberLongDenorm` columns must be kept in sync.

UpgradeCommission batch process is throwing an error when a reversal or payment plan refactor happens after a point in time transfer (BC-10917)

Issue – Under specific circumstances where a point in time transfer occurs, then a reversal or payment plan refactor occurs, then the BillingCenter database is upgraded to version 7.0.3, an error can later occur when running the `UpgradeCommission` batch process.

Workaround – Contact Guidewire Customer Support for assistance if you encounter this issue.

PolicyNumberLongDenorm not updated in upgrade (BC-10742)

Issue – Prior to 3.0, BillingCenter used 0-based term numbers for policies. Starting with 3.0, BillingCenter uses 1-based term numbers for policies to be consistent with PolicyCenter term numbering. If you are not upgrading from a version prior to 3.0 this issue does not affect you.

Because the `PolicyNumberLong` field (and the corresponding `PolicyNumberLongDenorm` field) can be a legal reference, BillingCenter does not change the value during upgrade.

The upgrade does renumber the `TermNumber` from being 0-based to being 1-based, since the `TermNumber` is not normally used as a legal reference. This behavior is to avoid a gap between term numbers. Otherwise, the terms would have a gap. For example, instead of `TermNumber` 0, 1, <upgrade> 3, 4, the upgrade renames the first two to make it 1, 2, 3, 4.

However, because the upgrade does not adjust `PolicyNumberLong` and `PolicyNumberLongDenorm`, there will be a gap in the `PolicyNumberLong` (for example 123456-0, 123456-1, <upgrade> 123456-3, 123456-4,. skipping any 123456-2 `PolicyNumberLong`).

Workaround – If you do not need to preserve your policy numbers after upgrade and you want your `PolicyNumberLong` and `PolicyNumberLongDenorm` values to match the new post-upgrade `TermNumber`, contact Guidewire Customer Support for information on how to update these fields.

IMPORTANT The `PolicyNumberLong` and `PolicyNumberLongDenorm` columns must be kept in sync.

Should not be able to edit Payment Plan after it has been used by a policy (BC-1726)

Issue – In the base configuration, it is possible to edit a Payment Plan even after it has been used by a policy. This is incorrect. Only the expiration date should be updateable.

Workaround – Configure the payment plan screen to only allow the expiration date to be updated.

Reversing Write-Off Commissions and Write-Off Commission Recovery in the UI (BC-7184)

Issue – The BillingCenter user interface does not provide a way reverse Write-Off Commissions and Write-Off Commission Recovery.

Workaround – Clone the Reverse button under the Payments tab and reverse `CmsnPayableReduction` and other transactions using the following action: `CurrentLocation.startEditing()`; `producerTransaction.reverse()`; and `CurrentLocation.commit()`.

Adding an existing contact following a new contact, causes the added contact not to display in the ListView on New Account and New Producer (BC-881)

Issue – Attempting to add an existing contact following a new contact causes the added contact to not display properly in the list view on the Add Contact wizard on the New Account and New Producer screens.

Workaround – Click on a column heading and the sort cause the screen to refresh and display the contacts correctly.

Rewrite BI with the same policynumber as the original policy is not incrementing the TermNumber and throws DBDuplicateKeyException (BC-2961)

Issue – Sending a rewrite billing instruction without specifying the term number will throw a `DBDuplicateKeyException` if term number “1” already exists for the policy. If you omit the term number, BillingCenter uses the number “1” for term number. If a policy period with the term number 1already exists then BillingCenter will throw an exception.

Workaround – Include a term number that is unique for the given policy number when sending a rewrite billing instruction.

Got exception when trying using the agency bill payment wizard to execute a promise with no dist items associated it (BC-4348)

Issue – Currently, if you click Execute on the Execute Agency Promise As Payment screen to apply a payment to a promise that has been pre-filled with zeroes, the following exception occurs:

`EvaluationException: Cannot re-add AgencyCyclePayment since it has already been removed.`

Workaround – Use the Payment Wizard to apply the payment instead.

AGBL statement Total Payment Applied is not correct after change 1 of 2 producer policies payment schedule (BC-2966)

Issue – The producer statement does not show the correct dollar amount after a change is made to one of the policy's payment plans on a policy with two producers after the statement was billed and partially paid.

Workaround –

Guidewire is aware of this issue.

Add NegativeWriteoff.ExecutionDate integrity check (BC-11699)

Issue – NegativeWriteoff.ExecutionDate is a nullable column, but it must be non-null for NegativeWriteoff records loaded from staging tables. The NegativeWriteoffLoaderCallback uses the ExecutionDate to populate the Transaction.TransactionDate. The integrity checks do not verify the ExecutionDate column is non-null.

Workaround – Guidewire is aware of this issue.

After PaymentPlan Refactor, Payments are not distributed correctly across commissions (BC-11892)

Issue – An issue occurs when performing a payment plan refactor on an agency bill policy that has been transferred. When redistributing agency bill payments, after payment plan change, BillingCenter calls the AgencyCycleDistPlugin plugin method `allocateForRedistribution`.

The base configuration of this plugin method uses the `gw.api.web.payment.ProRataRedistributionStrategy` class which does not reallocate commission fully in certain situations.

Workaround – The correct behavior is that, for each agency bill payment, the sum of commission amounts across all items targeting a specific charge should be equal before and after a payment plan change.

Contact Guidewire Customer Support for assistance in configuring the `allocateForRedistribution` method in the `AgencyCycleDistPlugin` plugin so that sum of commission amounts from each payment towards each charge matches the amounts contained in the allocation pool.

As a temporary alternative solution, if you encounter this problem, you can also manually redistribute the commission.

Cannot create New Collateral Charge with Money data type scale=2 appscale=0 (BC-10103)

Issue – When you define a data type with `scale=2` and `appscale=0`, you will receive the error message 'Exceeds the scale of 0' when you create a new collateral charge.

Workaround – Guidewire is aware of this issue.

Studio/Platform Issues

Studio Rules do not use correct capitalization for name of root object (PL-10740)

Issue – Rule set root objects are expected to begin with an upper-case letter and issues warning if a rule set root object does not.

Workaround – You can safely ignore the warning message.

User interface cannot handle starting multiple instances of a batch process (PL-12372)

Issue – The user interface cannot handle starting multiple instances of a batch process.

Workaround – To execute multiple instances of a batch process, start them from the command line. Also, to allow multiple instances to be run simultaneously, you must ensure that the `BatchProcess.isExclusive()` method returns `false`.

Type system refresh after PCF page title change does not update corresponding menu label (PL-13057)

Issue – The type system refresh after a PCF page title change does not update the corresponding menu label

Workaround – After updating a page title, restart the server to refresh menu labels and avoid null pointer exceptions due to stale references.

US-Locations.txt file with the US geodata from GreatData causes validation problems with United States Postal Service (USPS) data (PL-13384)

Issue – The US-Locations.txt file contains information that does not conform to United States Postal Service (USPS) standards for bulk mailings.

Workaround – The provided US-Locations.txt file is intended only for use in geocoding to identify addresses for a location. You can edit the US-Locations.txt file to conform to your particular address standards, and then import that version of the file instead.

GX model generated XSD cannot be parsed by JAXB (PL-13598)

Issue – XSD generated by the GX model cannot be parsed by JAXB.

Workaround – Add JAXB annotation elements to the XSD to specify the necessary metadata, such as class names, to enable JAXB to generate the Java class files. Contact Guidewire Support for a sample XSD file that is annotated this way.

Cannot make a field from a delegate into a localized column (PL-13761)

Issue – You cannot make a field from a delegate into a localized column.

Workaround – Move the column to be localized off the delegate and onto each of the implementing entities. Then, to make the column appear as though it exists on the delegate, define an enhancement property on the delegate that *delegates* to the appropriate column, depending on the implementing entity.

Problem with regen-java-api command and JAR files (PL-16351)

Issue – If you run the *BillingCenter/bin/gwpc regen-java-api* command, BillingCenter creates a *BillingCenter/java-api/lib* directory and puts JAR files for Java development in this directory. However, BillingCenter does not always perform this task correctly, resulting in some of the generated JAR files not being copied to this directory.

Workaround – For missing library files, copy or import into your Java development environment the missing library files from the following directory:

BillingCenter/admin/lib

Copy them into the following directory:

BillingCenter/java-api/lib

Renaming method or property throws ParseResultsException (PL-16633)

Issue – If you rename a property or a method or you change a method signature, and a workflow references that property or method in a Gosu field, BillingCenter throws *ParseResultsException*. This is the intended behavior.

Workaround – Restart the workflow engine. To do so:

1. Log into BillingCenter using an administrative account.
2. Access Internal Tools → Reload.

3. Click Reload Workflow Engine.

In some languages, web browsers render column headers of list views improperly (PL-18027)

Issue – In some languages, web browsers render some column headers of list views improperly if their column widths are specified too narrowly in their PCF definitions. For example, sometimes a numeric column is specified with a variable width of 1%. This narrow setting forces the browser to render the column too narrowly for the text of the translated column heading.

Workaround – Edit the PCF file that defines the column and clear the value from the width property. Without a specified value for the column width, browsers render the column widely enough to display the full text of the translated column heading.

Error on print for CheckBoxGroupInput Widget in Popup (PL-22209)

Issue – Problems occur whenever a user clicks a PrintToolbar button on a page that has a CheckBoxGroupInput widget.

Workaround – Alter the page that has the problem to remove the CheckBoxGroupInput widget.

Database upgrade does not handle nullable to non-nullable columns with a default value for subtypes (PL-23104)

Issue – For entity definitions, the automatic database upgrade converts nullable columns to non-nullable with a default value successfully. However, this column type conversion is not possible for columns in subtype definitions. BillingCenter implements non-nullable columns on subtype in the database as nullable because that column must have null values for rows that represents instances of other subtypes.

Workaround – Write a version trigger to populate the column with the default value for existing rows for the subtype. After you upgrade, BillingCenter enforces the column value to be non-nullable with the default value for new rows of the subtype.

New transport plugin definitions do not show in the list of valid transport plugins (PL-23317)

Issue – Newly implemented transport plugin definitions do not show in the list of valid transport plugins displayed by clicking the light-bulb icon next to the Transport Plugin field in the messaging destination editor.

Workaround – Restart Studio.

Find in Resources fails for resources under Data Model Extensions or Web Resources (PL-23320)

Issue – In Studio, the Find in Resources option does not work for resources that are in Data Model Extensions or Web Resources.

Workaround – To see the full resource path, hover over the resource name in the tab of the resource editor. Then, navigate the resource pane on the left to find the resource in the resource hierarchy.



chapter 25

Guidewire BillingCenter 7.0.3 Release Notes

Release 7.0.3

Document Revision: 02-July-2015

Copyright © 2001-2015 Guidewire Software, Inc. All rights reserved.

Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire Reinsurance Management, Guidewire ContactManager, Guidewire Vendor Data Management, Guidewire Client Data Management, Guidewire Rating Management, Guidewire InsuranceSuite, Guidewire ContactCenter, Guidewire Studio, Guidewire Product Designer, Guidewire Live, Guidewire DataHub, Guidewire InfoCenter, Guidewire Standard Reporting, Guidewire ExampleCenter, Guidewire Account Manager Portal, Guidewire Claim Portal, Guidewire Policyholder Portal, Gosu, Deliver Insurance Your Way, and the Guidewire logo are trademarks, service marks, or registered trademarks of Guidewire Software, Inc. in the United States and/or other countries.

All other trademarks are the property of their respective owners.

This material is confidential and proprietary to Guidewire and subject to the confidentiality terms in the applicable license agreement and/or separate nondisclosure agreement.

Guidewire products are protected by one or more United States patents.

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Support
- Issues and Major Changes
- Improvements and General Issues
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire BillingCenter.

Version Number

This release of Guidewire BillingCenter is 7.0.3

- For Guidewire Reporting, BillingCenter 7.0.3 supports IBM Cognos Business Intelligence 10.1 as the report server.
- If you are integrating with Guidewire PolicyCenter, this version of BillingCenter requires PolicyCenter 7.0.3 or later.

Installing This Release

Refer to the following:

- The *BillingCenter Installation Guide* for general installation information
- Prior BillingCenter release notes for any versions that you have skipped

Support

For assistance, visit the Guidewire Resource Portal – <http://guidewire.custhelp.com>

Issues and Major Changes

This section contains issues and major changes that may affect your installation.

- Upgrade Support
- Producer Statement Changes
- Commission Earning Changes
- Configurable Search
- Disaster Trouble Ticket API
- Base PCF File Changes
- Changes in This Release Provided in Upgrade Diff Report

Upgrade Support

Upgrading from BillingCenter 2.1.9 or newer versions is supported in this release.

See “Upgrading from 2.1.x” in the *Upgrade Guide* for more information.

IMPORTANT Contact Guidewire Customer Support to obtain SQL scripts you can run before and after your upgrade to create Database Upgrade Verification Reports. These reports help you in verifying that your data pre-upgrade correctly matches your data post-upgrade.

Producer Statement Changes

There are two new types of producer statements, **Upgrade** and **Pre-Upgrade**, which are generated only when upgrading from BillingCenter 2. **ProducerStatement** has a new type field. The **Producer Direct Bill Commission Statements** page has a new column displaying the producer statement type and includes special handling for the display of **Upgrade** statements.

- If you are not upgrading from BillingCenter 2, the **Type** column on the **Producer Direct Bill Commission Statements** page will always show **Production**. You can delete this column if you do not have legacy BillingCenter 2 data.
- If you are upgrading from BillingCenter 2, a set of **Upgrade** producer statements may be created during the upgrade process. Details of the new report and related changes follow.

Upgrade Producer Statement

BillingCenter now requires all earning and payable transactions to be associated with specific invoice items. Because this was not true in BillingCenter 2, the upgrade process must reverse all such legacy transactions and reissue them at the item level. The large number of resulting transactions are segregated on a separate ad-hoc **Upgrade** statement.

In order to identify these new types of producer statements, a new field has been added to **ProducerStatement**. The **ProducerStatement.Type** field has type **ProducerStatementType**. The **ProducerStatementType** typekey has three possible values: **Production**, **Pre-Upgrade**, and **Upgrade**. The default value for this field is **Production**. The meaning of each of these typekey values are as follows:

- **Pre-Upgrade** – Statements that were created and sent out in a BillingCenter 2 environment. These statements will remain unmodified, though all the earning and payable transactions attached to them will now be marked as reversed.
- **Upgrade** – Statements marked with the date the **UpgradeCommission** batch was run. These statements contain the reversal transactions of all previous earnings and payables on that Producer, as well as the new item-level earnings and payables required by BillingCenter 7. Because the reversals always have the opposite sign from the corresponding reissued transactions, these statements will have a total amount of \$0 in all cases, and exist solely to contain the activity generated by the commission upgrade.
- **Production** – All statements generated by the **ProducerPayment** batch process in BillingCenter 7 and later versions will be type **Production**. All new commission activity will be placed on one of these statements, and will be associated with invoice items.

Upgrade producer statements contain the set of the reversal and corresponding re-issuance transactions created during upgrade. Because the number of transactions can be very large, the **Producer Direct Bill Commission Statements** page hides the transactions when displaying **Upgrade** statements, and instead only shows the total number of transactions.

Migrating Producer Statements User Interface Configuration

When upgrading from BillingCenter 2, if you have custom configuration code in **ProducerStatement.pcf** or if you configured your own custom pages to display producer statements, note that the BillingCenter 7 producer statement user interface changes are important for performance and usability reasons.

You must merge the following changes with any custom configuration code that displays producer statements:

- **Upgrade** statements have line items hidden by default because the number of items may be very large.
- **Pre-Upgrade** statements use a different query to find legacy commission activity.

Commission Earning Changes

Commission earning is now more tightly tied to commission earning trigger events. Many commission payable criteria have been removed from the Commission Payable batch process, and BillingCenter now handles these earnings immediately when a trigger event occurs.

This is true for all commission plan Payable Criteria except date and custom criteria. There are three remaining criteria that must be handled by a batch process:

- Effective Date
- Expiration Date
- Custom

As a result of these changes, the commission payable batch process will now operate on only those three criteria. The commission payable batch process performance has been noticeably improved by these changes.

The payable criteria that earn immediately and have been removed from the commission payable batch process are the following:

- Payment Received
- Billing
- Binding
- Invoice Paid
- First Payment Received

Configurable Search

Before this release you could not use Gosu to configure search for types other than Account and Policy. Gosu search configuration has now been added for many more types.

WARNING Guidewire strongly recommends that you consider all the implications before customizing the **Search** tab. Adding new search criteria can result in significant performance impacts, particularly in large databases. Guidewire recommends that you thoroughly test any search customizations for performance issues before you move them into a production database.

Search for the following types now is now done in Gosu classes rather than in `search-config.xml`:

- Account
- AgencyBillMoneyRcvd
- Contact
- DelinquencyProcess
- DirectBillMoneyRcvd
- Invoice
- Payment
- Policy
- Producer
- SuspensePayment
- TroubleTicket

If you are upgrading and you have customized search for these types, you will have to migrate your customization to use these new classes. For assistance or information on how to skip this migration if you do not want to use Gosu search configuration, contact Guidewire Customer Support.

For information on configuring search for contacts, see “Searching for Contacts” in the *Contact Management Guide*.

For information on configuring search the above types other than Contact, see “Configuring BillingCenter Search” and “Working with Search Criteria in Gosu” in the *Configuration Guide*.

Disaster Trouble Ticket API

There is a new class, `gw.webservice.bc.bc700.TroubleTicketAPI`, which is exposed as a webservice. The methods of this class allow you to create *Disaster Trouble Tickets*. These are normal trouble tickets created all at once for a large number of policies and/or accounts associated with a set of postal codes.

This API may be useful in case of a catastrophe or disaster, if you need to place trouble ticket holds on all accounts and/or policy periods in a particular region. The `TroubleTicketAPI` class has two methods:

Create Disaster Trouble Holds on Accounts and Policies – The `ticket.createDisasterTroubleTicketsOnAccountsAndPoliciesWithPostalCodes` method creates trouble tickets for all accounts and policy periods with a specified set of postal codes. All hold types are set on the resulting trouble tickets.

Create Disaster Trouble Holds on Policies – The `createDisasterTroubleTicketsOnPoliciesWithPostalCodes` method creates trouble tickets for all policy periods with a specified set of postal codes. All hold types are set on the resulting trouble tickets.

Base PCF File Changes

All links below require the `readme_files` directory on your local disk.

BillingCenter release 7.0.2 to 7.0.3

- To view a report of the changes in the base PCF files in the `modules/bc` directory, [click here](#).
- To view a report of the changes in the base PCF files in the `modules/p1` directory, [click here](#).

Changes in This Release Provided in Upgrade Diff Report

Guidewire provides a report detailing certain differences between the current release and your prior release. This report describes changes in display keys, entities, typelists, and the Gosu API. To obtain your custom Upgrade Diff Report, visit the Guidewire Resource Portal.

Improvements and General Issues

BillingCenter Improvements and General Issues

The following are the improvements added to and the general issues corrected in this release:

ID	Description
Agency Bill	
BC-10054	Before this fix, BillingCenter allowed both of the following operations: <ul style="list-style-type: none">• A negative amount could be applied towards a distribution item targeting a positive invoice item.• A positive amount could be applied towards a distribution item targeting a negative invoice item. Neither of these are valid and BillingCenter now correctly does not allow either operation.
Commission	
BC-9085	Created an integrity check ensuring that <code>CmsnReserved.ItemEvent</code> is not loaded for a charge with a non-commissionable charge patterns.
BC-9814	If a the commission subplan for a policy period had 'Suspend for Delinquency' enabled, BillingCenter was suspending all commission earnings if and only if there were invoice items on the policy that are positive, not fully paid, and on 'Due' invoices. Now BillingCenter will suspend commissions when there is an active Delinquency Process associated with the PolicyPeriod. This makes the concept of when a policy is delinquent more consistent, and gives you more control over when commissions are suspended.

ID	Description
BC-10045	Fixed a problem which caused commission to incorrectly be increased if a commission override was done after a policy closure.
BC-10510	Added a consistency check to verify that there is no more than one active ItemCommission per PolicyRole per InvoiceItem.
BC-10569	The PrimaryCommissionAmount denormalized field on InvoiceItem was not correct in some cases. It was getting set to \$0 after a transfer on a non-primary role, or after a retroactive agency bill transfer. These cases are now handled correctly, and this field should always be correct.
Configuration	
BC-10805	<p>BillingCenter expects to always find a non-null value for the Type field on InvoiceItem.</p> <p>Some older releases of BillingCenter allowed the Type field to be empty. Gosu can also be used to create invoice items with Type set to null. This problem has been fixed by making InvoiceItem.Type non-nullable. Because it is non-nullable, it must have a default value. It is now set to <code>installment</code> by default.</p>
Consistency Checks	
BC-5828	<p>Added consistency checks to verify the following:</p> <ul style="list-style-type: none"> A Direct Bill Payment Item has one or zero payment item actions pointing to it where <code>PaymentItemAction.ActionKind = PaymentItemActionKind.Execute</code> A Direct Bill Payment Item has one or zero payment item actions pointing to it where <code>PaymentItemAction.ActionKind = PaymentItemActionKind.Reverse</code> and only if the item is reversed A Collateral Payment Item has one or zero payment item actions pointing to it where <code>PaymentItemAction.ActionKind = PaymentItemActionKind.Execute</code> A Collateral Payment Item has one or zero payment item actions pointing to it where <code>PaymentItemAction.ActionKind = PaymentItemActionKind.Reverse</code> and only if the item is reversed.
BC-9395	Added consistency checks to verify that PrimaryCommissionAmount, PrimaryCmsnPayableAmount, PrimaryPaidCommission, and PrimaryWrittenOffCommission match the corresponding fields on the active primary ItemCommission.
BC-9565	Added a consistency check to ensure that when <code>baseDistItem.PolicyPeriod</code> is null, <code>baseDistItem.InvoiceItem.Charge.TAccountContainer</code> is not a policy period T-account container.
BC-10413	<p>Before this fix, if the amount of a pending disbursement was set to zero, that disbursement would have its approval status left as Awaiting Approval but would have its associated approval activity marked as Completed. This would end up causing a consistency check failure because we have a consistency check which verifies that any Approvable (such as disbursement) with an approval status of Awaiting Approval has an open accompanying approval activity.</p> <p>This problem is fixed. When a pending disbursement has its amount set to zero BillingCenter will set the approval status to Rejected. As a result, disbursements in this state no longer cause a consistency check failure.</p>
BC-10707	Added consistency checks to verify that direct bill payments have a consistent account payer across all of the related transactions and line items.
BC-10714	Changed <code>InvoiceItemReversalDBCheckBuilder</code> database consistency check to filter out retired invoice items and charges in order to prevent false positives.
BC-10734, BC-10735	Added consistency checks for new Upgrade and Pre-Upgrade statements: <ul style="list-style-type: none"> Verifies there is not more than one Upgrade statement. Verifies there are no item-level producer transactions on Pre-Upgrade Statements. Verifies there are no charge-level earnings or payables (without invoice item) on Production Statements.
Contact Manager	
BC-10561	BC-10561 Updated the following ContactManagerSystemPlugin methods to correctly support multi byte character strings: <code>addContact</code> , <code>updateContact</code> , and <code>removeContact</code> .
Disbursement	

ID	Description
BC-9837	<p>There are two drop down filters on the Desktop Disbursements page that allow you to filter the disbursements by date and by status. Before this change, you had the option of selecting All in both of those filters. You could choose to show disbursements of All statuses and that were closed at any time. As a result you had the ability to display every disbursement in the system. To avoid performance problems this page has been updated so that the two filters no longer include an All option. You must now filter the disbursement results based on status and date (and no longer have the ability to see all disbursements in the system). In making this change, we had to make changes to DesktopDisbursements.pcf and DesktopDisbursementsLV.pcf and added two new supporting Java classes: DisplayDisbursementDateRange and DisplayDisbursementStatus.</p> <p>If you have modified these pcf files and want to include the performance improvement from these changes, you should merge these changes into your DesktopDisbursementsLV.pcf and DesktopDisbursements.pcf files. If you do not want this change, the supporting classes for the old configuration of the page are still in place so you do not have to not integrate these changes unless you want this performance improvement.</p>
Integration	
BC-10296	Made processing large numbers of billing instructions more efficient.
Internationalization	
BC-9447	<p>The Currency datatype has been modified to be BigDecimal instead of Number.</p> <p>This fixes errors that occurred for non-decimal currencies, such as Yen.</p>
Invoicing	
BC-10194	If a system crash occurred while processing a workItem for the Invoice batch process, the workItem would restart but fail, leaving the invoice with a billed status but without a correct snapshot.
BC-10366	BillingCenter groups invoices for processing. Before this fix, if an error occurred while processing one invoice, no further processing would be done on any invoices in that group. This fix allows BillingCenter to continue processing other accounts' invoices after an error which prevents one account's invoices to fail.
Logging	
BC-9780	The ChargeProRataTxn batch process will now log to debug instead of info to reduce the volume of log events generated.
Policy	
BC-9958, BC-10200	For customers using SQLServer, accessing a policy with no charges would cause an exception to be generated when attempting to view the policy summary page. This has been fixed.
BC-10201	Parsing of SQLServer queries with multiple comments were using a regular expression that treated the query as having one large comment (from the first opening-comment-token to the last closing-comment-token). The regular expression has been fixed to correctly match multiple comments.
Payment	
BC-10261	When the amount due in the first invoice is written off and a direct bill payment is made to pay all the remaining invoices in full, an exception would occur. This exception happened because at the time the first invoice was written off in full, the gross unsettled amount was set to zero. Creating a direct bill payment to pay this item, along with all the remaining invoice items, results in setting the amount to apply to zero for the item with gross unsettled amount of zero. It is a valid distribution and no longer causes an exception.
Performance	
BC-9959	The Transaction column which previously appeared on the AccountDBPaymentsLV and ChargesLV has been moved to an action button option in the first cell of the row.
BC-10207	Improved the performance of the ProducerPayment batch process.
Producer	
BC-9954	Improved the performance of the Producer Payment batch process.
Search	
BC-10666	Trouble ticket search was not returning any results for trouble tickets that are not associated with an account, policy or producer. Trouble ticket search results now include trouble tickets that are not associated with an account, policy or producer.

ID	Description
BC-10344	When searching using an invalid Policy number from the tab bar, an IncompatibleTypeException was thrown. This problem has been fixed.
BC-10481	Trouble ticket search, Account search, Agency Money Received search and Invoice search previously used validation on the zip code field that prevented partial zip codes from being used in a search. This has been fixed.

Platform Improvements and General Issues

ID	Description
Archiving	
PL-19035	Added a Download button to the (Server Tools) Archive Info page to more easily enable the collection and sending of archiving error information.
PL-20041	Fixed an issue in which the archiving schema (archiving.XSD) did not match the archiving output XML.
PL-20690	A null pointer exception sometimes occurred during restore of archived claims.
PL-21380	The regen-xsd command generated an XSD file with errors. The command has been updated to produce a valid XSD file.
Authentication	
PL-21737	The Gosu implementation of the AuthenticationSourceCreator plugin now supports JBoss.
Batch Processes	
PL-20755	Fixed an issue in which worker threads could unexpectedly quit if any database/JDBC error occurred during a time span longer than the maximum wait period.
Command Line Tools	
PL-19768	Fixed an issue that occurred while attempting to execute the verify-types command on a 64-bit JVM, which caused an Out of Memory error.
PL-20110	The command line utility gwbc regen-soap-api no longer fails on Japanese Windows.
Consistency Checker	
PL-16205	If you request a run of the database consistency checks using the <database> element attribute checker="true" (in config.xml), BillingCenter now prints a warning message in the log warning that this feature is deprecated. The message also instructs you to the Info Page or command line to run the consistency checks as a batch job.
Database	
PL-8729	Guidewire removed the configuration parameter TableEstimatePercent from config.xml. Instead, use the samplingpercentage attribute of the database statistics element in config.sys, as the following sample XML code shows. <databasestatistics samplingpercentage="20">
PL-10469	With this release, Guidewire applications do not start if the CurrentEncryptionPlugin parameter in config.xml specifies an encryption plugin implementation that does not exist.
PL-10468	Fixed an issue on the Database Storage Information page regarding the table display whenever switching from Index Physical Statistics to Tables and Indexes.
PL-14490	The data distribution tables now include a column for the internal major version number of the database schema.
PL-16977	Previously after a database upgrade, BillingCenter logged differences between the metadata configuration and the upgraded database schema at the INFO logging level. Now, BillingCenter logs differences at the WARN logging level.
PL-17742	On Oracle, changed database upgrade logic by providing additional optimizer hints to help improve upgrade performance.

ID	Description
PL-17835	Instead of evicting a database connection from the connection pool whenever an exception occurs, Guidewire applications now determine whether an exception was fatal to a connection before marking it for eviction. In addition, whenever the application marks a connection for eviction, the application logs the reason for the eviction. Constraint violations no longer cause evictions.
PL-20546	In data distribution tables, the size of numeric columns increased to hold larger values.
PL-20584	On Oracle, the Guidewire Profiler changed to help optimize the capture of the range of AWR snapshots.
PL-21145	Replaced the String value on DBNullConstraint Exception with display key Java.Database.DBException.NullConstraintViolation to enable localization of the message.
PL-21357	The Upgrade Info page and download now provide more consistent information about the steps that are being performed during an automated DB upgrade.
PL-21387	Corrected an issue that caused a null-pointer exception (NPE) in version triggers during a database upgrade.
PL-21448	On SQL Server, BillingCenter now provides additional information on database connections, which can then be captured and analyzed in the Server DMV Snapshot screen.
PL-21452	Corrected a issue that caused Document Management file downloads to fail if running over SSL in Internet Explorer 8 or earlier.
PL-21554	In previous versions on SQL Server, tuning queries was difficult because DBAs could not determine what part of the application generated the queries. Now, you can enable the new IdentifyQueriesViaComments parameter in config.xml to provide comments with contextual information in certain SQL Select statements sent to the relational database. The SQL comments are in the format: <i>ApplicationName:ActionName</i> <i>ApplicationName</i> is ClaimCenter, PolicyCenter, or BillingCenter. <i>ActionName</i> is the name of the page configuration file that submitted the SQL Select statement.
Database Support	
PL-19689	Guidewire has changed the mechanism for creating the database performance reports for Oracle and SQL Server. Instead of clicking on a button and waiting for the report to be generated and packaged in a ZIP file for immediate download, the report is now generated by a background batch process that stores the completed report in a new table named xx_dbperreport. This change requires a database upgrade to create this new table.
Database Upgrade	
PL-17616	You can now directly run SQL commands as part of custom database upgrade triggers. For details, see “Using the IDatamodelUpgrade Plugin” in the <i>BillingCenter Upgrade Guide</i> .
PL-20200	Guidewire has implemented optimization on the database upgrade. This optimization will improve the performance of new database creation, as well as database changes during minor and major upgrades.
PL-21310	Fixed an issue that caused database upgrades to fail if the WorkFlowWorkItem table was populated. Upgrade succeeded if the table was empty. The fix involved changing the WorkFlowWorkItem entity attribute from Final to Extendable.
PL-22007	In earlier versions, BillingCenter ran the AfterUpgradeVersionTrigger on fresh databases, which could cause problems. With this release, BillingCenter runs the AfterUpgradeVersionTrigger only on existing databases, not on fresh databases.
Document Management	
PL-18278	Guidewire added a new configuration parameter, AllowActiveXAutoInstall. This parameter controls whether BillingCenter automatically installs the Guidewire Document Assistant ActiveX control and supporting JScript support files used by the control. If AllowActiveXAutoInstall is set to false and AllowActiveX is true, BillingCenter does not install the control but uses the control if it has been manually installed. Contact Guidewire Support for information about manually installing the control. In addition, Guidewire added the ability to customize the location of JScript support files used by the control. See “Specifying Location for Guidewire Document Assistant Scripts” on page 94 in the <i>System Administration Guide</i> . Finally, Guidewire updated the version number of the control from 2.0.49/2.1.49 (white list) to 2.0.54/2.1.54 (white list).

ID	Description
PL-19958	Guidewire template documents using the Microsoft Word formats could not be generated if Microsoft Word was already open. The Guidewire Document Assistant ActiveX controls have changed versions from 2.0.49/2.1.49 (whitelist) to 2.0.52/2.1.52. Customers that manually deploy the control(s) will need to do so again.
PL-20275	Guidewire Document Assistant no longer adds extra section breaks to generated Microsoft Word documents.
PL-20549	Fixed a race condition in the initialization of hidden frames for BillingCenter that could cause the supporting JScript support files for the Document Assistant ActiveX control (TemplateRunner) to not be downloaded.
PL-20472	Previously, if errors occurred while saving a document, the user was not informed and changes were lost. Now, the user receives an error message.
Email	
PL-13582	A problem prevented files with Unicode file names attached to email from reaching email clients. Now, email clients that handle attachments with Unicode file names, such as Microsoft Outlook, receive the attachments from BillingCenter and send them correctly. If your email client does not handle Unicode file names, you must modify the EmailMessageTransport plugin to convert attachment file names to use Latin characters only.
Entities/Metadata	
PL-18068	Fixed a problem with verification of typelist typecodes on server startup. Typelist typecodes that match regardless of case fail validation. For example, the typecodes foobar and FooBar match disregarding case and thus fail validation.
PL-20127	Guidewire has corrected an issue with setting the appscale attribute for the Money data type. The appscale attribute worked for currencyamount columns, but not for money columns. The setting, specified in datatypes.xml, was being ignored when a column of type money was retrieved. This setting is now working. The appscale attribute controls the number of digits shown to the right of the decimal point for money or currency in single currency mode. It is similar to the scale attribute. The appscale attribute must be smaller than the scale setting. If defined, the appscale attribute overrides the scale attribute.
PL-20193	When an element of an owned array changes, the parent changes as well. For effdated owned arrays, the wrong parent changed. Now, the correct parent changes. This issue also affects effdated owned one-to-ones, because they are implemented as owned arrays.
PL-22597	In earlier versions of BillingCenter 7.x, if you defined an entity type that implements a Gosu interface, problems occurred if the interface had a method that returns an array of that entity type. None of the interface features were accessible on the entity that implemented the interface, and class cast exceptions could occur. Now, the interface features are accessible and class cast exceptions do not occur.
File Export	
PL-21546	Exported CSV files no longer contain UTF-8 BOM (byte order mark) characters at the beginnings of files that do not use UTF-8 encoding.
Global Cache	
PL-20332	A new configuration parameter, GlobalCacheDetailedStats, determines whether to collect detailed statistics for the global cache. Detailed statistics are data that BillingCenter collects to explain why items are evicted from the cache. Basic statistics, such as miss ratio, are still collected regardless of the value of this parameter. The GlobalCacheDetailedStats parameter is set to false by default. Set the parameter to true to help tune your cache. At runtime, use the Management Beans page to enable the collection of detailed statistics for the global cache. Whenever the GlobalCacheDetailedStats parameter is disabled, the Evict Information and Type of Cache Misses graphs are not displayed.
Gosu	
PL-20961	In Gosu you can now subclass an inner class that a supertype declares.
PL-22469	When an entity contains an implementsInterface element, that entity type is supposed to implement the interface specified in the implementsInterface element. Previously, methods inherited by the interface could be invoked, but references to the entity could not be assigned to variables of the interface type. This issue fixes the problem so that the entity type truly implements the interface, preserving assignability.
Messaging	

ID	Description
PL-13887	<p>IMPORTANT: The original version of the 7.0.3 release notes announced a new method <code>getDestinationStatus</code>. However, this method is not available in the product.</p> <p>The <code>IMessagingToolsAPI</code> has a new method to retrieve the status of a messaging destination. The method signature is:</p> <pre>getDestinationStatus(destID : int) : String</pre>
PL-18715	The Studio Messaging editor contained hard-coded strings for destination names, which made localization of destination names shown in the Admin tab difficult. Now, destination names are display keys. Therefore, destination names are included in the translatable resources of the application. Display keys for destination names are in the <code>Java.MessageDestination</code> level of the display key namespace.
Plugins	
PL-21409	In earlier versions, the User Authentication Service plugin (<code>AuthenticationServicePlugin</code>) threw four exceptions that <code>LoginForm.java</code> caught and displayed. There was no standard mechanism to add custom exceptions to the plugin and the login form. Now, you can modify the User Authentication Service plugin and throw <code>DisplayableLoginException</code> .
Staging Tables	
PL-20303	BillingCenter offers two new consistency checks: "One-to-one non-null check" and "Edge foreign key non-null check".
Studio IDE	
PL-19308	An issue occurred with Studio because some source control systems require a file or directory in a directory controlled by Studio. Studio interfered with the special, source control files. Now, Studio safely ignores third party files and directories required by some source control systems.
PL-20531	Fixed a problem that could cause Studio to not properly render selected text and caret positions in files containing true tab characters.
PL-21216	Studio now generates a warning message during PCF verification (Verify All) if the number of attempted verification passes exceeds the number set for Section Inclusion Limit in Tools → Verification Options → PCF Verification Options.
PL-21296	Modified the Studio PCF Editor so that the PCF Toolbox now shows subcategories for all Application customized widgets that you can use in widget filtering.
PL-21447	Corrected an issue in which the Studio Gosu editor periodically froze for a brief period of time on large files if you typed quickly.
Web – UI/Runtime	
PL-20150	Modified BillingCenter events processing logic to skip events for removed widgets. For example, if the values of two widgets have been changed in the same form submission, there will be two <code>onChange</code> events in the event queue. If the handler of the first event implicitly removes the widget that is the source of the second event, than the second event will no longer be processed (meaning that the associated <code>onChange</code> handler will not be invoked to avoid errors).
Web Services - WSI	
PL-18729	It is now possible to invoke local web services using the <code>wso.local</code> mechanism from Studio without a running server.
PL-20174	It is now possible to run WSI Web Services on a server other than the batch server in a cluster.
PL-21158	The Guidewire Profiler did not actually profile WS-I web services. Now whenever you select a WS-I web service to profile, BillingCenter provides profile information.
XML Element (and XSD types)	
PL-20076	Fixed an XML parsing bug that caused an <code>xs:choice</code> with a zero-width match to not be handled properly.

Known Issues and Limitations

This section describes known issues with this release of Guidewire BillingCenter.

- BillingCenter Known Issues

- Studio/Platform Issues

BillingCenter Known Issues

[Delinquency: Delinquency can be started on an account, even though there is no amount due, when negative charges are present \(BC-3746\)](#)

Issue – Delinquency can be started on an account when no amount is due if a negative charge on a policy is present.

Workaround – Invoke the `doDueTransaction()` method on the negative invoice item. This will execute a `ChargeDue` transaction for that invoice item which will prevent the problem.

Note that you can execute `invoiceItem.doDueTransaction()` method on a negative invoice item even if it is on an invoice that is in billed state not yet due. Later, when the invoice is made due, BillingCenter will just skip the negative invoice (because `invoiceItem.doDueTransaction()` already forced that invoice item to be due).

[Contact Manager: BC does not show updated Primary Address after CM contact Merge \(BC-9315\)](#)

Issue – If you are using BillingCenter with Contact Manager, there is a problem with merging contacts with different addresses. The primary address of the merged contact is not shown in BillingCenter.

Workaround – Guidewire is aware of this issue.

[Approve button on rejected Agency Bill Disbursement causes NullPointerException \(BC-9400\)](#)

Issue – You can click **Approve** on a rejected agency bill disbursement. This should not be allowed, and causes a null pointer exception.

Workaround – Disable the **Approve** button for agency bill disbursements. You can do this by modifying the `Available` property for the **Approve** toolbar button in `configuration\config\web\pcf\disbursement\DisbursementsLV.pcf` as follows.

Old:

```
selectedDisbursement.canApprove()
```

New:

```
selectedDisbursement.canApprove() && selectedDisbursement.Status != "Rejected"
```

[Commission: Commission override after a commission write-off causes write-off to be reserved \(BC-9536\)](#)

Issue – Several methods of the `BillingAPI` WS-I webservice, such as `issuePolicyPeriod()` and `cancelPolicyperiod()`, indirectly call `BillingInstructionUtil.executeAndCommit()` which commits the current bundle.

These methods also directly call `tryCatch()`, which commits a new bundle after executing the entire block of code inside of `tryCatch()`. As a result, the code tries to commit the same bundle twice.

Here are two ways you can encounter this problem:

1. If you try to create an activity within the block of code passed to `tryCatch()` after the call to execute the billing instruction, BillingCenter will throw a `DBAAlreadyExecutedException`.

For example:

```
function issuePolicyPeriod(issuePolicyInfo : IssuePolicyInfo, transactionId : String) : String{
    require(issuePolicyInfo, "issuePolicyInfo")
    var publicID = tryCatch( \ bundle -> {
        var pid= issuePolicyInfo.executeIssuanceBI()
        // code *here* to create a new activity
        return pid
    })
}
```

```

    }, transactionId)
    return publicID
}

```

- 2.** If you used `CreateTroubleTicketHelper` to create a trouble ticket in a billing instruction preupdate rule, you would again get a `DBA1readyExecutedException`. The following is an example of a billing instruction preupdate rule that would trigger the error:

```

// create trouble ticket with a delq hold until second invoice
var troubleTicketHelper = new CreateTroubleTicketHelper()
var ticket = troubleTicketHelper.createTroubleTicket((billingInstruction as
    PlcyBillingInstruction).PolicyPeriod)
ticket.Title = "Delinquency hold on issuance"
ticket.DetailedDescription = ticket.Title
ticket.Priority = "normal"
ticket.TicketType = "automatic"
ticket.Hold.setAppliedToHoldType(HoldType.TC_DELINQUENCY, true)
ticket.Hold.setReleaseDate(HoldType.TC_DELINQUENCY, gw.util.GWBaseDateEnhancement.CurrentDate)

```

Workaround –

- 1.** For this case, in the block of code passed to `tryCatch()`, do not create or update entities after the call to execute the billing instruction. For example:

```

function issuePolicyPeriod(issuePolicyInfo : IssuePolicyInfo, transactionId : String) : String{
    require(issuePolicyInfo, "issuePolicyInfo")
    var publicID = tryCatch( \ bundle -> {
        // code *here* to create a new activity
        var pid= issuePolicyInfo.executeIssuanceBI()
        return pid
    }, transactionId)
    return publicID
}

```

- 2.** For this case, instead of using `CreateTroubleTicketHelper` to create the trouble ticket, create related entities directly as in the following example in a PolicyPeriod enhancement:

```

var ticket = new TroubleTicket(this)
var ttJoinEntity = new TroubleTicketJoinEntity(ticket)
ticket.Title = ticketTitle
ticket.DetailedDescription = "Delinquency hold placed on Account with a " + ticket.Title
ticket.Priority = "normal"
ticket.TicketType = ticketType
ticket.Hold.setAppliedToHoldType( HoldType.TC_DELINQUENCY, true )
ticket.Hold.setReleaseDate(HoldType.TC_DELINQUENCY, releaseDate)
ticket.addToTroubleTicketJoinEntities(ttJoinEntity)
this.addToTroubleTicketJoinEntities(ttJoinEntity)
this.Account.addToTroubleTicketJoinEntities(ttJoinEntity)
ticket.Hold.checkForHoldAdditions()

```

PolicyCenter Integration: BC Renewal thread fails with exception on issue after removal of Alt Billing Account from Renewal in PC (BC-9540)

Issue – If you are integrated with PolicyCenter, you will encounter an error if you renew a policy that had an alternate billing account and you remove the alternate billing account in PolicyCenter prior to committing renewal.

Workaround – Guidewire is aware of this issue.

Commission: Commission override after a commission write-off causes write-off to be reserved (BC-9561)

Issue – Adding a commission override after a commission write-off causes the write-off to not be paid. This leads to mismatch between commission Reserve and Earned, payable and Paid.

Workaround – Guidewire is aware of this issue.

PolicyCenter Integration On Renewal Offer, Suspense Payment Should Be Applied At Policy Period (BC-9691)

Issue – When integrated with PolicyCenter, a suspense payment that is created with an offer number from a PolicyCenter renewal is getting applied to the owner account of the policy period. The suspense payment should be applied to the renewal policy period instead of the owner account.

Workaround – Guidewire is aware of this issue.

Payments: Reversing Producer Payments is throwing NullPointerException when the payment is related to a retired invoice item (BC-9749)

Issue – BillingCenter does not allow invoice items that have earned commission to be deleted. BillingCenter already prevents such items from being removed through the user interface. If an invoice item that has earned commission is deleted programmatically, errors will occur if the related payment item is reversed.

Workaround – Instead of deleting invoice items that have earned commission, set the invoice item amount to zero.

Disbursement: Changing the Disbursement reason from Automatic causes it to be sent out despite a hold (BC-9761)

Issue – The Disbursements batch process sends an disbursement if the disbursement reason is changed from 'Automatic' to something else, even if there is an active Trouble Ticket with a hold of type 'Disbursements' on the owning account. The BillingCenter user interface does not allow editing of automatic disbursements, but they can be edited through configuration. BillingCenter fails to account for this and the disbursement batch process treats a reason other than automatic to mean that it is a manual disbursement and so processes it, ignoring any holds.

Workaround – Use a custom extension to store more specific reasons for an automatic disbursement.

Payment Distribution: BC should not distribute payment amount that have been disbursed (BC-9783)

Issue – Reversing a DirectBillPayment is causing T-account Unapplied to become negative. This can happen when another direct bill payment on the same account gets modified.

Workaround – Create a recapture charge when unapplied goes below zero.

Changing Billing Method From AB to DB for earning on First Payment Received For Billed Statement Is Earning Commission (BC-10476)

Issue – Changing the billing method on a policy period from agency bill to direct bill with a producer that earns commission on First Payment Received will instead earn the commission when invoiced. Commission should not be earned until a payment is received.

Workaround – As part of transferring a policy period from agency bill to direct bill, add code to the policy period preupdate rule to reverse the earning transaction if the total amount paid on the policy period is zero. This will make the earning go back to zero for the policy period. The commission payable batch process will not identify these charge commissions (since their earning criteria will not match the selection criteria). When a real payment is posted to the policy, commission earning will be triggered by that event and be earned correctly.

PolicyNumberLongDenorm not updated in upgrade (BC-10742)

Issue – Prior to 3.0, BillingCenter used 0-based term numbers for policies. Starting with 3.0, BillingCenter uses 1-based term numbers for policies to be consistent with PolicyCenter term numbering. If you are not upgrading from a version prior to 3.0 this issue does not affect you.

Because the `PolicyNumberLong` field (and the corresponding `PolicyNumberLongDenorm` field) can be a legal reference, BillingCenter does not change the value during upgrade.

The upgrade does renumber the `TermNumber` from being 0-based to being 1-based, since the `TermNumber` is not normally used as a legal reference. This behavior is to avoid a gap between term numbers. Otherwise, the terms would have a gap. For example, instead of `TermNumber` 0, 1, <upgrade> 3, 4, the upgrade renames the first two to make it 1, 2, 3, 4.

However, because the upgrade does not adjust `PolicyNumberLong` and `PolicyNumberLongDenorm`, there will be a gap in the `PolicyNumberLong` (for example 123456-0, 123456-1, <upgrade> 123456-3, 123456-4, skipping any 123456-2 `PolicyNumberLong`).

Workaround – If you do not need to preserve your policy numbers after upgrade and you want your `PolicyNumberLong` and `PolicyNumberLongDenorm` values to match the new post-upgrade `TermNumber`, contact Guidewire Customer Support for information on how to update these fields.

IMPORTANT: The `PolicyNumberLong` and `PolicyNumberLongDenorm` columns must be kept in sync.

[UpgradeCommission batch process is throwing an error when a reversal or payment plan refactor happens after a point in time transfer \(BC-10917\)](#)

Issue – Under specific circumstances where a point in time transfer occurs, then a reversal or payment plan refactor occurs, then the BillingCenter database is upgraded to version 7.0.3, an error can later occur when running the `UpgradeCommission` batch process.

Workaround – Contact Guidewire Customer Support for assistance if you encounter this issue.

Studio/Platform Issues

[Studio Rules do not use correct capitalization for name of root object \(PL-10740\)](#)

Issue – Rule set root objects are expected to begin with an upper-case letter and issues warning if a rule set root object does not.

Workaround – You can safely ignore the warning message.

[User interface cannot handle starting multiple instances of a batch process \(PL-12372\)](#)

Issue – The user interface cannot handle starting multiple instances of a batch process.

Workaround – To execute multiple instances of a batch process, start them from the command line. Also, to allow multiple instances to be run simultaneously, you must ensure that the `BatchProcess.isExclusive()` method returns `false`.

['US-Locations.txt' file with the US geodata from GreatData causes validation problems with United States Postal Service \(USPS\) data \(PL-13384\)](#)

Issue – The `US-Locations.txt` file contains information that does not conform to United States Postal Service (USPS) standards for bulk mailings.

Workaround – The provided `US-Locations.txt` file is intended only for use in geocoding to identify addresses for a location. You can edit the `US-Locations.txt` file to conform to your particular address standards, and then import that version of the file instead.

[GX model generated XSD cannot be parsed by JAXB \(PL-13598\)](#)

Issue – XSD generated by the GX model cannot be parsed by JAXB.

Workaround – Add JAXB annotation elements to the XSD to specify the necessary metadata, such as class names, to enable JAXB to generate the Java class files. Contact Guidewire Support for a sample XSD file that is annotated this way.

Cannot make a field from a delegate into a localized column (PL-13761)

Issue – You cannot make a field from a delegate into a localized column.

Workaround – Move the column to be localized off the delegate and onto each of the implementing entities. Then, to make the column appear as though it exists on the delegate, define an enhancement property on the delegate that *delegates* to the appropriate column, depending on the implementing entity.

Problem with 'regen-java-api' command and JAR files (PL-16351)

Issue – If you run the `BillingCenter/bin/gwpc regen-java-api` command, BillingCenter creates a `BillingCenter/java-api/lib` directory and puts JAR files for Java development in this directory. However, BillingCenter does not always perform this task correctly, resulting in some of the generated JAR files not being copied to this directory.

Workaround – For missing library files, copy or import into your Java development environment the missing library files from the following directory:

`BillingCenter/admin/lib`

Copy them into the following directory:

`BillingCenter/java-api/lib`

Renaming method or property throws 'ParseResultsException' (PL-16633)

Issue – If you rename a property or a method or you change a method signature, and a workflow references that property or method in a Gosu field, BillingCenter throws `ParseResultsException`. This is the intended behavior.

Workaround – Restart the workflow engine. To do so:

1. Log into BillingCenter using an administrative account.
2. Access Internal Tools → Reload.
3. Click Reload Workflow Engine.

Error on print for CheckBoxGroupInput Widget in Popup (PL-22209)

Issue – Problems occur whenever a user clicks a `PrintToolbar` button on a page that has a `CheckBoxGroupInput` widget.

Workaround – Alter the page that has the problem to remove the `CheckBoxGroupInput` widget.

Unsupported config.xml FreeTextSearchEnabled parameter included in BC and CC (PL-23343)

Issue – The `config.xml` file includes the parameter `FreeTextSearchEnabled`. Free text search is a PolicyCenter only feature for this release. BillingCenter does not support Free text search and the `FreeTextSearchEnabled` should not be modified.

Workaround – Do not modify the default setting of `FreeTextSearchEnabled` (`value="false"`). Setting `FreeTextSearchEnabled` to "true" is not supported or recommended. If you enable this, you will not gain search functionality and you may encounter adverse behavior.



chapter 26

Guidewire BillingCenter 7.0.2 Release Notes

Release 7.0.2

Release Notes Update: 04-June-2012

IMPORTANT These release notes replace the release notes that were included in the official product release. Please disregard the earlier version of the release notes.

Document Revision: 02-July-2015

Copyright © 2001-2015 Guidewire Software, Inc. All rights reserved.

Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire Reinsurance Management, Guidewire ContactManager, Guidewire Vendor Data Management, Guidewire Client Data Management, Guidewire Rating Management, Guidewire InsuranceSuite, Guidewire ContactCenter, Guidewire Studio, Guidewire Product Designer, Guidewire Live, Guidewire DataHub, Guidewire InfoCenter, Guidewire Standard Reporting, Guidewire ExampleCenter, Guidewire Account Manager Portal, Guidewire Claim Portal, Guidewire Policyholder Portal, Gosu, Deliver Insurance Your Way, and the Guidewire logo are trademarks, service marks, or registered trademarks of Guidewire Software, Inc. in the United States and/or other countries.

All other trademarks are the property of their respective owners.

This material is confidential and proprietary to Guidewire and subject to the confidentiality terms in the applicable license agreement and/or separate nondisclosure agreement.

Guidewire products are protected by one or more United States patents.

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release

- Support
- Issues and Major Changes
- Improvements and General Issues
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire BillingCenter.

Version Number

This release of Guidewire BillingCenter is 7.0.2

- For Guidewire Reporting, BillingCenter 7.0.2 supports IBM Cognos Business Intelligence 10.1 as the report server.
- If you are integrating with Guidewire PolicyCenter, this version of BillingCenter requires PolicyCenter 7.0.3 or later.

Installing This Release

Refer to the following:

- The *BillingCenter Installation Guide* for general installation information
- Prior BillingCenter release notes for any versions that you have skipped

Support

For assistance, visit the Guidewire Resource Portal – <http://guidewire.custhelp.com>

Issues and Major Changes

This section contains issues and major changes that may affect your installation.

- Changes to the Logging API
- Base PCF File Changes
- Changes in This Release Provided in Upgrade Diff Report

Changes to the Logging API

Guidewire updated the logging API between BillingCenter 7.0.1 and 7.0.2. Guidewire has created a Knowledge Base article, *Changes to the logging API during upgrade to BillingCenter 7.0.2, ClaimCenter 7.0.2, and PolicyCenter 7.0.4 or later versions*, that describes changes to the logging API. Review this article if you are performing an upgrade, so that you can update your configuration to the new logging API. Contact Guidewire Support for a copy.

Base PCF File Changes

All links below require the `readme_files` directory on your local disk.

BillingCenter release 7.0.1 to 7.0.2

- To view a report of the changes in the base PCF files in the `modules/bc` directory, [click here](#).
- To view a report of the changes in the base PCF files in the `modules/p1` directory, [click here](#).

Changes in This Release Provided in Upgrade Diff Report

Guidewire provides a report detailing certain differences between the current release and your prior release. This report describes changes in display keys, entities, typelists, and the Gosu API. To obtain your custom Upgrade Diff Report, visit the Guidewire Resource Portal.

Improvements and General Issues

BillingCenter Improvements and General Issues

The following are the improvements added to and the general issues corrected in this release:

ID	Description
Agency Bill	
BC-9107	Fixed a bug in the context of making an Agency Bill payment and going to the Add Items search screen. If the user clicked on the Policy Period magnifying glass icon in order to search for a Policy Period, and then selected a Policy Period from the search results, the text value that was being populated in the Policy Period field when the user was taken back to the Add Items search screen was the Policy Period's display name, while the search was actually being done based on the Policy Period's policy number, resulting in no items being found. The search is now properly done on the Policy Period's display name (which is the same as the "policy number long" property).
Charges	
BC-9161	Fixed a problem where BillingCenter was failing to create negative subrogation charges, and also was failing to report that an error had occurred. It is now possible to create a negative subrogation charge.
Charge Holds	
BC-9198	Placing a hold on a charge that was causing an account to be delinquent did not exit delinquency. This problem has been fixed.
Commission	
BC-9156	Fixed a problem where if a commission override was done after policy closure then the commission adjustment amount would be incorrect. BillingCenter now calculates the commission adjustment amount correctly.
BC-9245	Policy period level commission was being calculated on non-commissionable charges when the commission rate had an override. This problem has been fixed.
Configuration	
BC-9262	An unused copy of <code>PaymentRequestDV.pcf</code> was removed from the <code>bc-configuration</code> directory.
BC-9521	The <code>ParentAccount</code> edge foreign key field on the <code>Account</code> entity had previously been specified with <code>loadable=false</code> . As a result this field could not be populated by staging table loads. BillingCenter has changed the definition of this field to have <code>loadable=true</code> so that it can now be populated by staging table loads.
BC-9747	Increased the visibility of the <code>entity.InvoiceItem.canRemove()</code> method so it can be called from Gosu code.
Consistency Checks	
BC-6285	Added a consistency check for approvables that are not either executed or rejected and have no open approval activity. It is possible to create these approvables programmatically, but they can not be created through the user interface.

ID	Description
BC-9499	BillingCenter has a consistency check which checks whether an agency bill payment item is pointing at the correct policy period. In this case, the correct policy period is the policy period corresponding to the invoice item that is being paid by the payment item. This consistency check has been updated so that it also performs the same check against direct bill payment items.
Contact Manager	
BC-9679	Fixed a problem that would occur when a new contact was sent to BillingCenter from PolicyCenter when ContactManager was also integrated with those two applications. The problem was an exception that occurred in ContactManager because of the data BillingCenter was sending about the contact received from PolicyCenter. BillingCenter now sends the correct data to ContactManager, and the exception no longer occurs.
Document Management Integration	
BC-9531	Previously the set methods for Account, Policy and Producer foreign key fields on Document were not available. This meant that customers were not able to set those foreign keys through Gosu. These set methods have now been exposed and customers will now be able to set the values of these fields through Gosu.
Internationalization	
BC-1063	The currency and data format are now localized in the 'Funds Tracking -> Use of Funds -> Payment Item Group Details' screen.
BC-4595	Before this fix the names of T-account owners (like Account, PolicyPeriod, Producer) were displayed as unlocalized text in the Journal and Ledger screens. So they would show up as English no matter what language the user had chosen. That has been fixed so that now they will show up as localized strings that will be displayed appropriately based on the user locale.
BC-9111	The CommissionRate format was not being updated when the language was changed. This has been fixed.
BC-9196	T-account names can be localized on the Ledger page by modifying chargepatterns.xml.
BC-9218	T-account names in the Transactions tab are now localizable.
BC-9242	Updated the date format in strings created by the PCPolicySystemPlugin to use the correct international format.
BC-9269	Fixed a bug where using a browser with a different locale than the locale of the server would cause Javascript to incorrectly parse amounts with thousands and decimal separators when editing an agency bill payment distribution.
BC-9377	There is a screen in the Create Disbursement wizard where a Date was not formatted for the user locale. That problem has been fixed and the date will now be correctly formatted for the current user locale.
BC-9399	There was a hard coded usage of 12:01 in PCPolicySystemPlugin (used when integrating PolicyCenter with BillingCenter). This caused problems when the customer ran with certain locales. This has been fixed.
BC-9447	The Currency datatype has been modified to be BigDecimal instead of Number. This fixes errors that occurred when the default currency was not decimal.
BC-9582	The Agency Bill Payment Wizard was assuming default currency was decimal. For currencies such as Yen, this was not true and could cause problems. The Agency Bill Payment Wizard now correctly rounds currency amounts to the smallest unit appropriate for the default system currency.
BC-9596	The method PolicyPeriodEquityDatingAspect.getAmountGivenDay() has been updated to respect the default currency rather than assuming a decimal currency.
BC-9598	Fixed a problem where BillingCenter assumed a decimal currency. This would have caused problems for customers who use non-decimal currencies (such as Yen). BillingCenter now checks the system default currency to determine whether a decimal currency is in use.
BC-9599	Fixed a problem with currency globalization in BillingCenter. There were some usages of 0.00 for zero money in transaction code. This could cause problems for customers that use a non-fractional default currency.
BC-9659	Before this fix, the Money data type was not connected to a field validator. The Money data type is now connected to the Money field validator and the field validator has been updated. Customers can merge these changes into their datatype.xml and fieldvalidator.xml files.

ID	Description
BC-9661	Updated AgencyBillExceptionsLV.pcf to use display keys instead of strings so that the AgencyBillExceptionsLV.pcf can be localized
Invoicing	
BC-9130	Before this fix, if there was a gross write-off amount on an invoice item that we were reversing, we were erroneously copying that amount onto the reversal, resulting in incorrect past due amounts. There is also a data fix for this issue. Contact Guidewire support for more information.
BC-9634	BillingCenter groups invoices for processing. Before this fix, if an error occurred while processing one invoice, no further processing would be done on any invoices in that group. This fix allows BillingCenter to continue processing other accounts' invoices after an error which prevents one account's invoices to fail.
Payment Instrument	
BC-9460	Default (immutable) payment instruments created by BillingCenter once on initialization are now always created in the same order when BillingCenter is started with an empty database.
Performance	
BC-4434	For funds tracking purposes, BillingCenter copies reversed payment items from a direct bill payment being modified into the latest version of that direct bill payment. Before this fix, Billing Center was also doing this unnecessarily for agency bill payments (which are not included in funds tracking) and this was causing performance problems when doing repeated edits on a large agency bill payment. We've fixed this problem by updating BillingCenter to not copy over reversed payment items for agency bill payments.
BC-9139	Improved performance of query used in ProducerImpl.calculateCommissionTotal().
BC-9386	Changed how BillingCenter builds queries for the invoice billed and invoice due batch processes, resulting in better performance.
BC-9546	Improved performance of Producer Summary Screen. Commission Paid Year to Date on the Producer Summary screen has been optimized and should render about twice as quickly now.
PolicyCenter Integration	
BC-9151	This fix addresses a problem that occurred when BillingCenter executed a renewal billing instruction. If the offer number was set, BillingCenter would try to reverse all suspense payments with the same offer number, even if they were already reversed.
BC-9325	Fixed a problem with <code>gw.webservice.policycenter.bc700.RenewalInfoEnhancement.executeRenewalBI()</code> where BillingCenter was not copying the offer number field to the new policy period after receiving the policy renewal from PolicyCenter. This prevented the renewal suspense payment from being applied. BillingCenter now captures the renewal offer number from PolicyCenter and correctly applies the suspense payment once the renewal is confirmed.
BC-9632	Fixed two problems with PolicyCenter integration: When renewing or rewriting a policy period in PolicyCenter, the renewal or rewrite would always use the prior policy period's producer code. If a different producer code was chosen for the renewal or rewrite, it was ignored. When a new renewal policy period was created by PolicyCenter, BillingCenter would set the BillingCenter policy period's producer code to null. If the new renewal specified a producer, it was ignored by BillingCenter.
Search	
BC-9155	Fixed a bug where a "too many parameters" SQL exception would be thrown when accessing an account that has thousands of policies.
BC-9518	The Gosu search utility class <code>gw.search.PolicySearchCriteria</code> was inadvertently hidden in the BillingCenter 7.0.1 release. This has been fixed, and the <code>PolicySearchCriteria</code> class is exposed for customer use or modification.

ID	Description
Upgrade	
BC-9203	<p>Fixed the following problem. BillingCenter was creating payment instruments only for the currently selected Payment Method on the Account.</p> <p>BillingCenter will now create a payment instrument if the payment method details for ACH or Credit Card are set for an account, regardless of whether or not the currently selected payment method is ACH or Credit Card.</p> <p>Note: If the currently selected payment method for an account is misc, we will create a payment instrument for misc, but as misc has no payment method details fields by default, we cannot know that there should be a misc payment instrument unless it is currently selected.</p> <p>Also, because ACH and Wire have the same payment method details fields, there is no way to know whether BillingCenter should create an ACH or a Wire payment instrument. BillingCenter will always arbitrarily select ACH.</p>

Platform Improvements and General Issues

Activities	
PL-19637	Fixed an issue that caused the Activity Escalation work queue writer to create duplicate work items.
Batch Process	
PL-18232	It is now possible to extend any WorkQueue subclass exposed in BillingCenter.
PL-18404	<p>Removed the BatchServer config parameter from config.xml. Instead, use the JVM argument or registry element <code>isBatchServer</code> to set the batch server.</p> <p>Also, added the ability to promote a non-batch server to be the batch server on the ClusterInfo page. If a non-batch server is found in the cluster, you can click PromoteToBatch for that server to make that particular server the batch server.</p> <p>Also, added the <code>promoteToBatch</code> operation to the Cluster Management Bean.</p>
PL-18637	<p>The default BillingCenter behavior for query builder result sets is to retrieve all entries from the database into the application server. If the result set is quite large, this can cause problems. Now, you can use the following method to set the query retrieve chunk size.</p> <pre>public final void setChunkingByID(IQueryResult queryResult, int chunkSize)</pre>
Clustering	
PL-11554	<p>Modified how Guidewire applications discover and maintain the batch server in the clustered environment. Prior to this change, the batch server was discovered with the help of a message exchange through the JGroups cluster communication channel. Users starting up their nodes at approximately the same time could end up with 2+ batch servers in the cluster, if cluster was not completely formed yet.</p> <p>This changes uses the underlying database to discover and to keep track of the batch server currently active in the cluster.</p>
PL-19141	Improved the usability of (Server Tools) Cluster Info page. The page now shows Server ID and host name information. Guidewire also changed the menu slightly.
Cognos Integration	
PL-19226	The <code>bc cognos</code> command now builds an application-specific Cognos integration ZIP file, with a name of the form <code>bc-cognos-reporting-integration.zip</code> . Each zip file contains application-specific integration scripts (<code>bc-config</code> and <code>bc-import</code>). These scripts no longer require the namespace/application parameter.
Command Line Tools	
PL-13663	Removed hard-coded Java heap settings that could cause out-of-memory issues while running certain tools.
PL-18225	Added WSI (document literal) webservices support to Administration command-line tools.
PL-18856	Fixed an issue that involved regenerating the Data Dictionary. Running <code>gwbc regen-dictionary</code> did not actually produce the Data Dictionary files.

PL-18892	<p>Guidewire has added the following optional argument to the data dictionary generation tool:</p> <pre>maxSPVInclusions</pre> <p>This value of this parameter defines the depth for second pass verification that limits the number of shared sections that are included the verification of PCF types using verify-all. For example:</p> <pre>gwbc regen-dictionary -DmaxSPVInclusions=1000</pre> <p>For this case, the second pass compilation of PCF files stops after 1000 permutation of modal PCF files.</p> <p>Note</p> <ul style="list-style-type: none"> If you do not include this option, the dictionary generation tool behaves as before. Only use positive integer values for the maxSPVInclusions property.
Configuration Upgrade	
PL-12804	Fixed an issue in which the upgrade tool did not handle rules upgrade properly.
Consistency Checker	
PL-16199	One-to-one relationships can have at most one entity on each side of the relationship. Under certain conditions, Guidewire is unable to create a unique index to enforce this. In those cases, Guidewire now creates a consistency check to verify the integrity of the data.
Contact Domain	
PL-18245	Fixed an issue with IgnoreProperty used in contact-sync-config.xml that caused it to take affect on all subtypes of Contact, rather than only those specified. This functionality now works properly and allows fields to be ignored on contact synchronization only for the specified subtypes.
Database	
PL-1541	<p>BillingCenter now explicitly names primary key constraints using the following conventions:</p> <ul style="list-style-type: none"> Oracle – Primary key constraint on the table PX_CONTACT will be named as PK_CONTACT. All other databases – Primary key constraint on the table PX_CONTACT will be named as PX_CONTACT_PK.
PL-17234	BillingCenter no longer creates shadow tables (tables whose names starts with XXt_ and XXt_t_) for new databases if you start the application server in dev mode (JVM parameter -Dgw.server.mode=dev). BillingCenter now creates shadow tables only if you set the server.running.tests system property to true (explicitly or programmatically).
PL-17286	Guidewire is providing a newer version of the Oracle JDBC driver in the Guidewire application distribution. If you are using an external connection pool instead of the Guidewire-bundled connection pool, then you must copy the new driver from the distribution to the location from which the application server data source will pick up the Oracle JDBC driver.
PL-17618	BillingCenter now logs the reason for generating a certain step during a database upgrade to the server console log (at the DEBUG level). This change also adds this information to the (Server Tools) Upgrade Info page as well.
PL-18749	<p>With this release, whenever a database upgrade begins, the upgrade process marks the database with the time the upgrade started and the host and batch server from which it began. At the time the database upgrade succeeds, the upgrade process removes the marker. If a second database upgrade begins before the first upgrade finishes, the second upgrade process detects the marker from the first upgrade process and throws an exception.</p> <p>A second upgrade process can begin if the first upgrade fails and you attempt a restart without first restoring the database. Alternatively, a second upgrade process can begin if a second batch server begins a database upgrade before the first upgrade process completes successfully.</p>
PL-19421	Fixed an issue on SQL Server that caused a stack trace overflow if both of the following were true:
	<ul style="list-style-type: none"> There was a connection problem during application start-up. Configuration parameter MigrateToLargeIDsAndDatetime2 was set to false.
Document Management	
PL-18360	A new version of the Guidewire Document Assistant ActiveX control will be downloaded to the client browser.
PL-18432	Improved performance for retrieval of extremely large documents through the ActiveX Guidewire Document Assistant.

PL-18653	Guidewire has upgraded the BFO PDF library to version 2.11.20. Refer to the following web site for more details on the changes in this version: http://bfo.com/viewtext.jsp?url=products/pdf/docs/CHANGELOG.txt&title=Big+Faceless+PDF+Library+Changelog
PL-18704	Microsoft Word field forms (text fill-in field) are now restored correctly while generating a document from an Microsoft Word Template.
Entities/Metadata	
PL-14982	Modified the Studio Tools → Verify... feature so that it now reports errors/warnings during verification of Entity Names.
PL-18162	Guidewire has added a validator that prevents you from attempting to denormalize a localized column. Guidewire does not support localized columns for search denorm columns.
PL-18266	Modified the entity type loader to dynamically generate interfaces for the following: <ul style="list-style-type: none"> • Entities that do not have a class • Entities that are extended with a delegate or Java interface that the compile-time backing class does not extend <p>The type loader now uses this runtime interface to create array instances. The runtime <code>impl</code> class now implements this dynamically-generated interface as well.</p>
PL-18469	Added a new attribute to entity types called <code>typelistTableName</code> . Use this attribute for non-final entities to specify the name of the corresponding subtype typelist table. If not specified, then BillingCenter uses the name of the entity as the subtype typelist table name. This capability is useful if an entity name is too long to become a valid typelist table name.
PL-19655	Fixed a problem with columns defined as <code>oneToOne</code> that disregarded <code>nullOk = false</code> . Now, BillingCenter prevents you from committing empty <code>oneToOne</code> columns that do not allow nulls.
PL-20127	Corrected an issue with setting the <code>appscale</code> attribute for <code>MoneyDataType</code> . The feature worked for <code>currencyamount</code> columns, but not for <code>money</code> columns. The setting, specified in <code>datatypes.xml</code> , was being ignored if a column of type <code>money</code> was retrieved. This setting is now working. The <code>appscale</code> attribute controls the number of digits shown to the right of the decimal point for money or currency in single currency mode. It is similar to the <code>scale</code> attribute. It must be smaller than the <code>scale</code> setting and overrides the <code>scale</code> attribute if defined.
Geocoding/Proximity Search	
PL-1461	Two parameters are added to the plugin registry for the Geocode plugin. The <code>geocodeDirectionsCulture</code> parameter specifies the locale for geocoded addresses and routing instructions returned from a geocoding and routing service. For example, use the locale code <code>ja-JP</code> for addresses and instructions for Japan. The <code>imageryCulture</code> parameter specifies the language for map imagery. For example, use the language code <code>ja</code> for maps labeled in Japanese.
PL-18619	Proximity search works around some bad execution plans by disabling index fast full scan and hash join if executing on Oracle. New configuration parameters <code>DisableIndexFastFullScanForProximitySearch</code> and <code>DisableHashJoinForProximitySearch</code> control the workaround. The default value for these parameters is <code>false</code> . These parameters have no effect on databases other than Oracle.
Global Cache	
PL-18322	Fixed an issue with method <code>ProximitySearchQueryUtils.filterWithinRadiusLatLong</code> that did not properly return all the requested points. This issue affected radius searches with a smaller radius value than the number of desired results returned.
Internal Tools/Server Tools Pages	
PL-18017	Guidewire has made the Internal Tools tab available in test mode.
PL-18839	Guidewire has significantly improved the performance of the GW Profiler.
PL-18845	Modified the (Server Tools) Batch Process Info page so that you can now download detailed records for any particular batch process for a specific date range. Clicking Download now opens a page in which you can specify the date range to download for a given batch process.
Localization	
PL-18490	Modified <code>localization.xml</code> to let the <code>thousandsSymbol</code> attribute of the <code><NumberFormat></code> element type accept a non-breaking space character as the thousands separator. For example: <code><NumberFormat thousandsSymbol="&#160;></code>
Logging	

PL-14722	<p>Added the following new configuration parameters that configure application logging:</p> <ul style="list-style-type: none"> • LoggerCategorySource • LoggersShowLog4j
PL-14764	<p>Added the UserInterface.Performance logging category for tracing and debugging issues in the user interface. You can configure the logging category to include the ID of the user.</p>
PL-18745	<p>Added support for two new Log4j MDC (Mapped Diagnostic Contexts) keys to include information about the current user in log messages. To use them, include a sequence conforming to the following string in your <code>ConversionPattern</code> in <code>logging.properties</code>:</p> <pre>%-<LL>.<HH>X{user userName userID}</pre> <p>The parameters have the following meanings:</p> <ul style="list-style-type: none"> • <LL> – The minimum size of the field. If the actual value is shorter, the user name gets padded with spaces on the right. • <HH> – The maximum size of the field. If the actual value is longer, the user name gets truncated from the left. • user – Prints the user's internal numeric ID number (for example, 4231341234) (This was available in previous releases and remains unchanged for compatibility.) • userName – Prints the user's real-world name (for example, John Smith). • userID – Prints the user's username in the system (for example, jsmith). <p>For example:</p> <pre>%-16.16X{userName}</pre>
PL-20907	<p>Guidewire updated the logging API between the following releases:</p> <ul style="list-style-type: none"> • BillingCenter 7.0.1 and 7.0.2 • ClaimCenter 7.0.1 and 7.0.2 • PolicyCenter 7.0.3 and 7.0.4 <p>Guidewire has created the following Knowledge Base article that describes the changes made to the logging API:</p> <p><i>Changes to the logging API during upgrade to BillingCenter 7.0.2, ClaimCenter 7.0.2, and PolicyCenter 7.0.4 or later versions</i></p> <p>Review this article if you are performing an upgrade, in order to update your configuration to the new logging API. If necessary, contact Guidewire Support for a copy.</p>
Messaging	
PL-18678	<p>The <code>MessageSenderRunnable.run</code> method now includes the <code>messageID</code> in the log. The log entry will look similar to the following:</p> <pre>MW.MessageSenderRunnable.run (dest destination id): Entering run() for messageId message id</pre>
PL-19136	<p>Fixed an issue in which a race condition in a non-clustered server could lead to inconsistent results and/or runtime exceptions when modifying entities within a message transport.</p>
PL-19149	<p>After locking out a user who has reached the maximum allowed login failures, BillingCenter no longer ignores generated events. Instead, BillingCenter executes the transaction with a service token having a super user. This means that BillingCenter now generates <code>UserChanged</code> events properly.</p>
Other	
PL-3760	<p>Guidewire has made the following modification to <code>config.xml</code>:</p> <ul style="list-style-type: none"> • Removed element <code><security sessiontimeoutsecs="10800"/></code> • Added configuration parameter <code><param name="SessionTimeoutSecs" value="10800"/></code> as its replacement <p>BillingCenter performs an automatic upgrade of <code>config.xml</code> for this change.</p>

PL-15462	<p>Guidewire applications contain several configuration parameters that accept a comma-separated list of values.</p> <p>Now, BillingCenter parses a comma-separated list of values according to CSV format and trims each value automatically. This permits you to use spaces (and tabs, new lines) for more readability while specifying such a value.</p>
	<p>For example:</p> <ul style="list-style-type: none"> • A value that before the fix must be without spaces – AUTO, PR, GL, TRAV • The same value after the fix can use spaces – AUTO, PR, GL, TRAV
	<p>The following is a more technical description of the format:</p> <ul style="list-style-type: none"> • Fields must be separated with commas. • Leading and trailing spaces are ignored—unless the field is delimited with double-quotes. In that case the white spaces are preserved. • Embedded comma – You must delimit the field with double-quotes, with the comma inside the double quotes. • Embedded double-quotes – You must double embedded double-quote characters and you must delimit the field with double-quotes. • Embedded line-breaks – You must surround the field with double-quotes. • Always Delimiting - You can always delimit a field with double-quotes. If not strictly needed, the delimiters will be parsed and discarded by the reading applications.
PL-15914	<p>Guidewire has added Save and Cancel buttons on the (Server Tools) Management Beans → Guidewire Managed Bean Properties page that you can use to save or cancel a change to a bean property value. The buttons become active after you edit an editable property.</p>
PL-19311	<p>Guidewire has upgraded the Joda-Time third-party library version to 2.0.</p>
Queries	
PL-17541	<p>Modified GWDBFunctionEnhancement.gsx and added support for DBFunction calls of MIN and MAX on java.util.Date.</p>
PL-19590	<p>Fixed an issue that caused an error in the BillingCenter user interface by a query that contained a column definition twice.</p>
Studio IDE	
PL-13352	<p>Fixed an issue in which Studio ignored the formatting settings for String literals set in Studio Tools → Options → Colors and Fonts.</p>
PL-15219	<p>Fixed an issue with PCF TemplateInputWidget in which autocomplete inserted a value that appeared to be correct, but which failed validation.</p>
PL-15887	<p>Fixed an incorrect message that Studio showed in the Verify Result pane if verifying a valid entity name in the Entity Names editor.</p>
PL-18203	<p>Fixed an issue in which the Studio debugger step-over functionality stopped on Guidewire internal code in between executing Rule Condition and Rule Action code.</p>
PL-18671	<p>Guidewire has modified Studio behavior in regards to readonly mode:</p>
	<ul style="list-style-type: none"> • Studio now displays a padlock button on the status bar that is visible only if it is in readonly mode. If you click the button, Studio displays a modal message box indicating the reason(s) why it is in readonly mode. • Studio disables the Save button any time that it is in readonly mode. • Studio changes the Save button tooltip in readonly mode to display the reason that save is not active in this mode. This is the same message that Studio shows if you click the padlock icon on the status bar.
PL-18905	<p>Fixed an issue in which it was possible to edit configuration files even if Studio was in readonly mode.</p>
PL-19466	<p>Fixed a problem that occurred when you dragged a rule to the top of a rule set. Guidewire Studio incorrectly overwrote the first rule instead of inserting the lower order rule above it. Now Studio inserts the lower order rule correctly above the top rule.</p>
PL-19510	<p>Fixed an issue that caused errors in the Guidewire Studio Find in Path functionality when searching rule resources if the resources contained files that were not proper Gosu rule resources. Proper Gosu rule resources contain Gosu code and end with the file extensions .grs or .gr. Now, Find in Path displays non-Gosu rule resources as errors without affecting search results for valid resources.</p>
PL-19568	<p>Guidewire Studio now supports 64-bit Java Virtual Machines.</p>
Utilities	

PL-11681	<p>The Gosu StringUtil class used many Perl 5, version 3, regular-expression, syntax-compatible functions to search for and replace strings. These functions were implemented by the Apache/Jakarta ORO library, which has been retired. See http://jakarta.apache.org/oro/ for details.</p> <p>The same functionality for regular-expression, syntax-compatible search and replace of strings is now available in the <code>java.util.regex</code> package and the <code>java.lang.String</code> class.</p> <p>Before this change, you used the following coding pattern:</p> <pre>StringUtil.replace(inputString, "s/" + regexString + "/" + replacementString + "/g")</pre> <p>Now, instead, you use the following coding pattern:</p> <pre>var pattern = java.util.regex.Pattern.compile(regexString) var patchedString = pattern.matcher(inputString).replaceAll(replacementString)</pre> <p>Simpler search and replace functions are also available, such as:</p> <pre>java.lang.String.replace(String, String).</pre>
PL-15733	Guidewire has updated the Google Guava library to release 10.
Web – UI/Runtime	
PL-15548	Printing a second-level list view no longer results in missing items or items to print more than once.
PL-18303	Removed errors caused while adding multiple entries in list views for custom entity types.
PL-19454	Fixed an issue that occurred while using Internet Explorer 9 in which clicking a drop-down list did not render the list properly.
PL-19602	Added the ability to retain the scroll position after clicking a pull-right menu item. This fix adds the <code>retainScrollPosition</code> property to the <code>MenuItem</code> widgets.
Web Services	
PL-18361	BillingCenter now provides the ability to implement a pre-existing WSDL in Gosu for WSI web service.
PL-18450	BillingCenter now consumes MTOM-enabled WS-I web services. MTOM is the W3C Message Transmission Optimization Mechanism that efficiently sends binary data to and from Web services. You do not need to do anything to take advantage of this new feature.
PL-18589	Added a logging category for WSI webservices to log each request, <code>XML.Request</code> . Each request will be logged at the debug level. This include the connecting address and user, if available, as well as the request qname (which is unique for each operation).
Work Queues	
PL-18675	Fixed several issues that involved the (Server Tools) Work Queue Info page.
PL-19003	You can now start or stop work queues from <code>IMaintenanceToolsAPI</code> .
PL-19004	Added the attribute <code>orphansFirst</code> to a worker thereby making it possible to specify that BillingCenter is to process orphans before new items on this worker.
PL-19059	The (Server Tools) Work Queue Info page now reports the last time that BillingCenter processed a work item, showing either the last success time or the last exception time. If it was an exception, the Info page also reports the number of consecutive work items that resulted in an exception. The page shows two other dates as well. The last notification time is the last time the worker woke up. If it found work, BillingCenter updates the last batch found time.

Known Issues and Limitations

This section describes known issues with this release of Guidewire BillingCenter.

- BillingCenter Known Issues
- Studio/Platform Issues

BillingCenter Known Issues

Delinquency: Delinquency can be started on an account, even though there is no amount due, when negative charges are present (BC-3746)

Issue – Delinquency can be started on an account when no amount is due if a negative charge on a policy is present.

Workaround – Invoke the `doDueTransaction()` method on the negative invoice item. This will execute a `ChargeDue` transaction for that invoice item which will prevent the problem.

Note that you can execute `invoiceItem.doDueTransaction()` method on a negative invoice item even if it is on an invoice that is in billed state not yet due. Later, when the invoice is made due, BillingCenter will just skip the negative invoice (because `invoiceItem.doDueTransaction()` already forced that invoice item to be due).

PolicyCenter Integration: BC Renewal thread fails with exception on issue after removal of Alt Billing Account from Renewal in PC (BC-9540)

Issue – If you are integrated with PolicyCenter, you will encounter an error if you renew a policy that had an alternate billing account and you remove the alternate billing account in PolicyCenter prior to committing renewal.

Workaround – Guidewire is aware of this issue.

Contact Manager: BC does not show updated Primary Address after CM contact Merge (BC-9315)

Issue – If you are using BillingCenter with Contact Manager, there is a problem with merging contacts with different addresses. The primary address of the merged contact is not shown in BillingCenter.

Workaround – Guidewire is aware of this issue.

Commission: Commission override after a commission writeoff causes writeoff to be reserved (BC-9536)

Issue – Several methods of the `BillingAPI` WS-I webservice, such as `issuePolicyPeriod()` and `cancelPolicyperiod()`, indirectly call `BillingInstructionUtil.executeAndCommit()` which commits the current bundle.

These methods also directly call `tryCatch()`, which commits a new bundle after executing the entire block of code inside of `tryCatch()`. As a result, the code tries to commit the same bundle twice.

Here are two ways you can encounter this problem:

1. If you try to create an activity within the block of code passed to `tryCatch()` after the call to execute the billing instruction, BillingCenter will throw a `DBA1readyExecutedException`.

For example:

```
function issuePolicyPeriod(issuePolicyInfo : IssuePolicyInfo, transactionId : String) : String{
    require(issuePolicyInfo, "issuePolicyInfo")
    var publicID = tryCatch( \ bundle -> {
        var pid= issuePolicyInfo.executeIssuanceBI()
        // code *here* to create a new activity
        return pid
    }, transactionId)
    return publicID
}
```

2. If you used `CreateTroubleTicketHelper` to create a trouble ticket in a billing instruction preupdate rule, you would again get a `DBA1readyExecutedException`. The following is an example of a billing instruction preupdate rule that would trigger the error:

```
// create trouble ticket with a delq hold until second invoice
var troubleTicketHelper = new CreateTroubleTicketHelper()
```

```
var ticket = troubleTicketHelper.createTroubleTicket((billingInstruction as  
    PlcyBillingInstruction).PolicyPeriod)  
ticket.Title = "Delinquency hold on issuance"  
ticket.DetailedDescription = ticket.Title  
ticket.Priority = "normal"  
ticket.TicketType = "automatic"  
ticket.Hold.setAppliedToHoldType(HoldType.TC_DELINQUENCY, true)  
ticket.Hold.setReleaseDate(HoldType.TC_DELINQUENCY, gw.util.GWBaseDateEnhancement.CurrentDate)
```

Workaround –

1. For this case, in the block of code passed to `tryCatch()`, do not create or update entities after the call to execute the billing instruction. For example:

```
function issuePolicyPeriod(issuePolicyInfo : IssuePolicyInfo, transactionId : String) : String{  
    require(issuePolicyInfo, "issuePolicyInfo")  
    var publicID = tryCatch( \ bundle -> {  
        // code *here* to create a new activity  
        var pid= issuePolicyInfo.executeIssuanceBI()  
        return pid  
    }, transactionId)  
    return publicID  
}
```

2. For this case, instead of using `CreateTroubleTicketHelper` to create the trouble ticket, create related entities directly as in the following example in a PolicyPeriod enhancement:

```
var ticket = new TroubleTicket(this)  
var ttJoinEntity = new TroubleTicketJoinEntity(ticket)  
ticket.Title = ticketTitle  
ticket.DetailedDescription = "Delinquency hold placed on Account with a " + ticket.Title  
ticket.Priority = "normal"  
ticket.TicketType = ticketType  
ticket.Hold.setAppliedToHoldType( HoldType.TC_DELINQUENCY, true )  
ticket.Hold.setReleaseDate(HoldType.TC_DELINQUENCY, releaseDate)  
ticket.addToTroubleTicketJoinEntities(ttJoinEntity)  
this.addToTroubleTicketJoinEntities(ttJoinEntity)  
this.Account.addToTroubleTicketJoinEntities(ttJoinEntity)  
ticket.Hold.checkForHoldAdditions()
```

Commission: Commission override after a commission writeoff causes writeoff to be reserved (BC-9561)

Issue – Adding a commission override after a commission write-off causes the write-off to not be paid. This leads to mismatch between commission Reserve and Earned, payable and Paid.

Workaround – Guidewire is aware of this issue.

PolicyCenter Integration On Renewal Offer, Suspense Payment Should Be Applied At Policy Period (BC-9691)

Issue – When integrated with PolicyCenter, a suspense payment that is created with an offer number from a PolicyCenter renewal is getting applied to the owner account of the policy period. The suspense payment should be applied to the renewal policy period instead of the owner account.

Workaround – Guidewire is aware of this issue.

Commission: Commission payable batch process does not pay the commission when an exception is cleared with past due invoice (BC-9714)

Issue – The commission payable batch process is not handling cleared exceptions on past due invoices correctly. The commission amount should be moved to commission payable from commission reserved, but remains in the commission reserved T-account.

Workaround – Guidewire is aware of this issue.

Disbursement: Changing the Disbursement reason from Automatic causes it to be sent out despite a hold (BC-9761)

Issue – The Disbursements batch process sends an disbursement if the disbursement reason is changed from 'Automatic' to something else, even if there is an active Trouble Ticket with a hold of type 'Disbursements' on the owning account. The BillingCenter user interface does not allow editing of automatic disbursements, but they can be edited through configuration. BillingCenter fails to account for this and the disbursement batch process treats a reason other than automatic to mean that it is a manual disbursement and so processes it, ignoring any holds.

Workaround – Use a custom extension to store more specific reasons for an automatic disbursement.

Payment Distribution: BC should not distribute payment amount that have been disbursed (BC-9783)

Issue – Reversing a DirectBillPayment is causing T-account Unapplied to become negative. This can happen when another direct bill payment on the same account gets modified.

Workaround – Create a recapture charge when unapplied goes below zero.

Payments: Reversing Producer Payments is throwing NullPointerException when the payment is related to a retired invoice item (BC-9749)

Issue – BillingCenter does not allow invoice items that have earned commission to be deleted. BillingCenter already prevents such items from being removed through the user interface. If an invoice item that has earned commission is deleted programmatically, errors will occur if the related payment item is reversed.

Workaround – Instead of deleting invoice items that have earned commission, set the invoice item amount to zero.

Internationalization: Journal/Ledger - TAccountOwnerType is not getting translated to Japanese (BC-9902)

Issue – Before this release the names of T-account owners (such as Account, PolicyPeriod, Producer) were displayed as unlocalized text in the Journal and Ledger screens. As a result they would not be displayed in the selected user language. These strings have been extracted and now can be correctly localized, but the following code is necessary.

Workaround – Create a record in display.properties. For example:

```
entity.PolicyPeriod = PolicyPeriod
```

Agency Bill: BC is not able to distribute negative amount against positive payment item (BC-9924)

Issue – BillingCenter does not correctly distribute a negative payment amount against positive invoice item.

Workaround – Guidewire is aware of this issue.

Studio/Platform Issues

Build command 'verify-types' fails for 64-bit JVM (PL-19768)

Issue – Executing the verify-types command results in the following error on 64-bit versions of Java:
`java.lang.OutOfMemoryError`

Workaround – Use a 32-bit version of Java.

First time you click the arrow of the typekey input, the drop-down menu does not open (PL-10134)

Issue – The drop-down menu does not open on the first click of the arrow on a typekey input. Instead, the help text opens.

Workaround – Turn off help text on focus by setting `InputHelpTextOnFocus` to `false` in `config.xml`. After you do that, the help text shows only if you mouse over the input, and it does not interfere with opening a drop-down menu.

Studio Rules do not use correct capitalization for root object's name (PL-10740)

Issue – Rule set root objects are expected to begin with an upper-case letter and issues warning if a rule set root object does not.

Workaround – You can safely ignore the warning message.

User interface cannot handle starting multiple instances of a batch process (PL-12372)

Issue – The user interface cannot handle starting multiple instances of a batch process.

Workaround – To execute multiple instances of a batch process, start them from the command line. Also, to allow multiple instances to be run simultaneously, you must ensure that the `BatchProcess.isExclusive()` method returns `false`.

Type system refresh after PCF page title change does not update corresponding menu label (PL-13057)

Issue – The type system refresh after a PCF page title change does not update the corresponding menu label.

Workaround – After updating a page title, restart the server to refresh menu labels and avoid null pointer exceptions due to stale references.

US-Locations.txt file with the US geodata from GreatData has special characters that cause validation problems with United States Postal Service (USPS) data (PL-13384)

Issue – The `US-Locations.txt` file contains information that does not conform to United States Postal Service (USPS) standards for bulk mailings.

Workaround – The provided `US-Locations.txt` file is intended only for use in geocoding to identify addresses for a location. You can edit the `US-Locations.txt` file to conform to your particular address standards, and then import that version of the file instead.

GX models that reference virtual fields and enhancements throw null pointers if null (PL-13560)

Issue – GX models that reference virtual fields and enhancements throw null pointers when these fields and enhancements are null. While object path expressions are generally null safe, they are not null safe when the expression navigates to a virtual field or enhancement.

Workaround – Include null checks and error handling to prevent referenced virtual fields or enhancements that are null from causing null pointer exceptions. For example, the following object path navigates to a virtual field.

```
anActivity.ActivityStatus
```

Use the null-safe period operator instead of the standard operator, as the following example shows:

```
anActivity?.ActivityStatus
```

Sending email with file attachment with unicode filename is not correctly handed over to the mail server (PL-13582)

Issue – An email with a file attachment that has a unicode file name is not sent to the mail server correctly.

Workaround – Use Latin characters for file names on attached files.

[GX model generated XSD cannot be parsed by JAXB \(PL-13598\)](#)

Issue – XSD generated by the GX model cannot be parsed by JAXB.

Workaround – Add JAXB annotation elements to the XSD to specify the necessary metadata, such as class names, to enable JAXB to generate the Java class files. Contact Guidewire Support for a sample XSD file that is annotated this way.

[Cannot make a field from a delegate into a localized column \(PL-13761\)](#)

Issue – You cannot make a field from a delegate into a localized column.

Workaround – Move the column to be localized off the delegate and onto each of the implementing entities. Then, to make the column appear as though it exists on the delegate, define an enhancement property on the delegate that *delegates* to the appropriate column, depending on the implementing entity.

[Problem with 'regen-java-api' command and JAR files \(PL-16351\)](#)

Issue – If you run the *BillingCenter/bin/gwbc regen-java-api* command, BillingCenter creates a *BillingCenter/java-api/lib* directory and puts JAR files for Java development in this directory. However, BillingCenter does not always perform this task correctly, resulting in some of the generated JAR files not being copied to this directory.

Workaround – For missing library files, copy or import into your Java development environment the missing library files from the following directory:

BillingCenter/admin/lib

Copy them into the following directory:

BillingCenter/java-api/lib

[Renaming method or property throws 'ParseResultsException' \(PL-16633\)](#)

Issue – If you rename a property or a method or you change a method signature, and a workflow references that property or method in a Gosu field, BillingCenter throws *ParseResultsException*. This is the intended behavior.

Workaround – Restart the workflow engine. To do so:

1. Log into BillingCenter using an administrative account.
2. Access Internal Tools → Reload.
3. Click Reload Workflow Engine.

[Studio does not properly render tab characters \(PL-20531\)](#)

Issue – Studio does not properly render selected text and caret positions in files containing true tab characters. This typically occurs with code that you paste into Studio from another IDE such as Eclipse.

Workaround – Start Studio with the following parameter:

studio -Dstudio_disable_improved_font_rendering=true

[Specifying env attribute in <database> generates run time exception \(PL-20815\)](#)

Issue – If you do not define a default *env* specifier for the *<database>* element in *config.xml*, Studio generates the following exception and fails to start:

```
[java] java.lang.RuntimeException: java.lang.RuntimeException: No appropriate database found in
configuration: env = 'null'
[java] at com.guidewire.studio.shell.StudioDependenciesManager
```

```
.startDependencies(StudioDependenciesManager.java:201)
```

Workaround – Do one of the following:

- Provide a default env in the database registry section of config.xml.
- Set a default env by adding the following to the command to start Studio:

```
studio -Dgw.cc.env
```




chapter 27

Guidewire BillingCenter 7.0.1 Release Notes

Release 7.0.1

Document Revision: 02-July-2015

Copyright © 2001-2015 Guidewire Software, Inc. All rights reserved.

Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire Reinsurance Management, Guidewire ContactManager, Guidewire Vendor Data Management, Guidewire Client Data Management, Guidewire Rating Management, Guidewire InsuranceSuite, Guidewire ContactCenter, Guidewire Studio, Guidewire Product Designer, Guidewire Live, Guidewire DataHub, Guidewire InfoCenter, Guidewire Standard Reporting, Guidewire ExampleCenter, Guidewire Account Manager Portal, Guidewire Claim Portal, Guidewire Policyholder Portal, Gosu, Deliver Insurance Your Way, and the Guidewire logo are trademarks, service marks, or registered trademarks of Guidewire Software, Inc. in the United States and/or other countries.

All other trademarks are the property of their respective owners.

This material is confidential and proprietary to Guidewire and subject to the confidentiality terms in the applicable license agreement and/or separate nondisclosure agreement.

Guidewire products are protected by one or more United States patents.

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in This Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire BillingCenter.

Version Number

This release of Guidewire BillingCenter is 7.0.1

- For Standard Reporting, BillingCenter 7.0 supports IBM Cognos Business Intelligence 10.1 as the report server.

If you are integrating with Guidewire PolicyCenter, this version of BillingCenter requires PolicyCenter 7.0.1 or later.

Support

For assistance, visit the Guidewire Resource Portal – <http://guidewire.custhelp.com>

Introducing This Release

Refer to the following:

- The *BillingCenter Installation Guide* for general installation information
- The *BillingCenter Reporting Guide* if you plan to use the integrated reporting feature
- BillingCenter release notes previous releases for any versions that have been skipped

Changes in This Release

This section describes the product changes in this release.

- Base PCF File Changes
- Changes in This Release Provided in Upgrade Diff Report

Base PCF File Changes

All links below require the `readme_files` directory on your local disk.

BillingCenter release 7.0.0 to 7.0.1

- To view a report of the changes in the base PCF files in the `modules/bc` directory, [click here](#).
- To view a report of the changes in the base PCF files in the `modules/p1` directory, [click here](#).

Changes in This Release Provided in Upgrade Diff Report

Guidewire provides a report detailing certain differences between the current release and your prior release. This report describes changes in display keys, entities, typelists, and the Gosu API. To obtain your custom Upgrade Diff Report, visit the Guidewire Resource Portal.

BillingCenter Improvements and General Issues

The following are the improvements added to and the general issues corrected in this release:

ID	Description
Account	
BC-1038	Fixed an issue where, when an account is in edit mode, an incorrect error message was displayed when an invalid day of month was entered for the 'Invoice Fixed On' day on the account Summary screen.
Accounting Subsystem	

ID	Description
BC-4074	Prior to this fix, BillingCenter would allow unexecuted billing instructions to be created. This fix prevents that from occurring, except for PremiumReportDueDate billing instructions.
BC-9037	Added data security by preventing changing the amount of a charge after it has already been submitted to BillingCenter. The amount can still be modified before its billing instruction is executed, but not after.
Commission	
BC-262	<p>Fixed a problem that occurred if you did a payment plan refactor after transferring commission on an individual invoice item. After the payment plan refactor, the item commission would not match the T-account commission balances. The default item commissions would claim to have all the commission reserve, but the non-default producer code that you transferred to in the first step would still have the transactions for the commission in its reserve T-account.</p> <p>Commission is now transferred to the correct T-account in this scenario.</p> <p>To fix data created before this fix:</p> <ol style="list-style-type: none"> 1. Identify data that needs to be fixed. Instructions follow for finding all charges where item commissions values do not match the T-account values on their charge commissions. 2. Call <code>redistributeAllCommission()</code> on each of these charges. <p>Instructions for identifying data that needs to be fixed:</p> <p>Look for charges with charge commissions that break the following consistency checks in <code>ItemCommissionDenormFieldsDBCheckBuilder</code>:</p> <ul style="list-style-type: none"> • ChargeCommission Payable balance should equal the sum of its <code>ItemCommission.CmsnPayableAmount</code> • ChargeCommission Reserve balance should equal the sum of (<code>CommissionAmount - CmsnPayableAmount - WrittenOffCommission - PaidCommission</code>) for each <code>ItemCommission</code> • ChargeCommission write-off balance should equal the sum of the <code>WrittenOffCommission</code> for each <code>ItemCommission</code> • Sum of ChargeCommission Expense balance and the <code>CmsnsExpenseRollup</code> transactions should equal the sum of <code>CommissionAmount</code> for each <code>ItemCommission</code>
BC-443	<p>Fixed a problem where a point-in-time transfer followed by a payment plan refactor could result in incorrect commission being earned by the old and new producer.</p> <p>Now the reversed invoice items do not allocate any commission for inactive producers. The commission for the active producer is properly removed or reversed, and distributed across the new invoice items. Additionally, as a result of this change, we now only reverse commission write-offs against the <i>active</i> producer when reversing an invoice item.</p>
BC-4458	ZeroCmsnEarnedMarkers are now correctly removed during retroactive transfer and payment item reversal, allowing future earnings for those invoice items.
BC-8897	The method <code>com.guidewire.bc.domain.invoice.InvoiceItem.getExecutedPaymentItems()</code> has been removed. Calls to <code>getExecutedPaymentItems()</code> should be replaced with calls to <code>getActivePaymentItems()</code> .
BC-8923	The ReservedCmsmEarned consistency check was updated to be consistent with changes made in BillingCenter 7.0. A ProducerContext with an on payment received PayableCriteria is now valid if it has either a <code>DistItem</code> or a <code>ChargeWrittenOff</code> associated with it.
BC-9045	<p>Fixed two cases where the earned commission of inactive producers was being modified inappropriately:</p> <ul style="list-style-type: none"> • Manual add/remove invoice items to a charge • Payment Plan Refactor with Planned Only <p>Now any items that were fully earned on an inactive producer are left alone, and commission is only redistributed to the active producer.</p>
Consistency Check	
BC-8979	Improved the performance of several consistency checks.
BC-9010	<p>Two new consistency checks were added:</p> <ul style="list-style-type: none"> • Ensure that if a <code>BaseDistItem</code> is executed, its <code>BaseDist</code> is distributed. • Ensure that if a <code>BaseDistItem</code> is not executed, its <code>BaseDist</code> is not distributed. <p>Note that reversed <code>BaseDistItems</code> are not checked.</p>
BC-9021	When transferring an agency bill policy, BillingCenter now transfers any commission on the remainder invoice item. This prevents a discrepancy between the payer and earner for the remainder item.

ID	Description
BC-9022	<p>Fixed a problem where a policy transfer sometimes caused a consistency check failure, because the end date of a policy commission might be milliseconds after the start date for the next policy commission. The problem was addressed with these two changes:</p> <ul style="list-style-type: none"> BillingCenter base product will no longer produce any overlap between policy commissions. The restriction in the consistency check was relaxed, allowing up to 5 seconds of overlap between producers. This ensures that old data will not break the consistency check.
ContactManager Integration	
BC-3527	Fixed a problem where accounts and policies sent to BillingCenter from PolicyCenter were not successfully created in BillingCenter if the associated contact had multiple addresses. The account or policy is now correctly created.
Delinquency	
BC-8918	Fixed a problem where the delinquency process would not assign anything if round robin was selected. Using round robin assignment as part of the delinquency process now works correctly.
Funds Tracking	
BC-8898	A single <code>PaymentPlanRefactor</code> <code>PaymentItemNarrative</code> context is now created when a payment plan refactor is executed.
BC-8908	Fixed a problem with funds tracking where <code>PaymentItemsNarrative.Context.descendantActions()</code> was returning the complete set of actions for every context in the payment item narrative tree. <code>PaymentItemNarrative.Context.descendantActions()</code> now correctly returns only those actions that are directly or indirectly in the current context.
Invoice	
BC-8709	Fixed a problem that could occur when creating a new invoice stream. If the periodicity was changed after overriding the invoice day, then an error would be displayed on the screen. That error no longer occurs.
BC-8920	When BillingCenter makes an invoice that includes negative items billed or due, the negative items are distributed as credits as follows: <ul style="list-style-type: none"> BillingCenter will use these credits to pay invoice items in order from last to first, by invoice item event date BillingCenter will limit distribution to the amount outstanding (amount billed or due), regardless of the account's distribution limit All other types of payment allocation and distribution (including from T-account Unapplied) remain unchanged
BC-9132	Fixed a problem that could occur when reversing an invoice item that has a gross written-off amount. We were incorrectly copying the denormalized gross written off amount to the reversal, and thus making reversal payments with incorrect amounts. BillingCenter now handles this situation correctly. There is a data fix for this issue. To repair incorrect data call <code>FixUtil.fixCorruptedDataFromReversingInvoiceItemsWithGrossWriteOffs(bundle)</code> .
Performance	
BC-121	<code>ItemEvent</code> is no longer retrievable. Removing an <code>ItemEvent</code> will no longer retire it. Instead, removing an <code>ItemEvent</code> will now delete it from the database. Upgrading to 7.0.1 will delete retired <code>ItemEvent</code> entities from the database.
BC-893	<code>LineItem</code> is no longer retrievable. <code>LineItems</code> should never be retired or removed. If there happen to be retired <code>LineItems</code> prior to upgrading to 7.0.1, they will be deleted from the database. Before this change, removing a <code>LineItem</code> would retire it. Now removing a <code>LineItem</code> will delete it from the database.
BC-9086	BillingCenter improved the performance of several loader callbacks.
PolicyCenter Integration	
BC-7925	BillingCenter now has an additional typekey in the default LOBCode typelist. The new typekey has code <code>HomeOwners</code> and name <code>Home Owners</code> , which corresponds to the same typekey in the PolicyCenter LOBCode typelist.

ID	Description
BC-9053	<p>Fixed a problem where some tools such as Altova XMLSpy and soapUI failed to generate class files from the BillingAPI WSDL.</p> <p>BillingAPI has new methods, so if you used another tool to generate class files from the BillingAPI WSDL, you must regenerate your class files.</p>
BC-9314	<p>Fixed a problem with <code>gw.webservice.policycenter.bc700.RenewalInfoEnhancement.executeRenewalBI()</code> where BillingCenter was not copying the offer number field to the new policy period after receiving the policy renewal from PolicyCenter. This prevented the renewal suspense payment from being applied.</p> <p>BillingCenter now captures the renewal offer number from PolicyCenter and correctly applies the suspense payment after the renewal is confirmed.</p>
Reporting	
BC-8922	<p>Two fields were added to BillingCenter: <code>PolicyPeriod.TermConfirmed</code> and <code>Charge.WrittenDate</code>. These fields are not used by any internal BillingCenter processes. They have been created to support reporting requirements. Policy administration systems, including Guidewire PolicyCenter, can optionally set these fields so that reports have this information.</p>
BC-9061	<p>A new entity, <code>ReportingGroup</code>, was added to BillingCenter. This optional entity was created for reporting purposes. It has three properties: <code>Jurisdiction</code>, <code>UWCompany</code>, and <code>Product</code>. Certain entities have a foreign key to <code>ReportingGroup</code>, so a <code>ReportingGroup</code> can be assigned to those entities. Those entities include <code>Account</code>, <code>Producer</code>, <code>BaseMoneyReceived</code>, <code>Disbursement</code>, <code>ProducerPayment</code>, and <code>SuspensePayment</code>.</p>
Search	
BC-7862	<p>You can now configure account and policy period searches by modifying the new Gosu classes <code>AccountSearchCriteria</code> and <code>PolicySearchCriteria</code>.</p> <p>Account and policy period search was previously implemented in Java, which could not be modified and configured.</p>
Staging Table	
BC-8846	<p>BillingCenter was updated to ensure that commission reserve <code>ItemEvents</code> are not loaded against non-commissionable charges or billing instructions. Two changes were made:</p> <ul style="list-style-type: none"> Added an integrity check ensuring that if <code>CmsnReserved</code> <code>ItemEvents</code> are not linked to charges on billing instructions we don't load those transactions. Added code to check each billing instruction and ensure that <code>OnsetChargeBI</code> and <code>OffsetChargeBI</code> are not loaded into staging tables.
BC-8847	<p>Added an integrity check to ensure that if a <code>PolicyCommission</code> is set to <code>DefaultForPolicy = true</code>, each <code>ChargeCommission</code> pointing to it has an <code>ItemCommission</code> for each <code>InvoiceItem</code> on its <code>Charge</code>.</p>
BC-9099	<p>The <code>ChargeCommission</code> field on <code>ChargePaidFromProdCtx</code> is now properly populated when loading from staging tables.</p>

ID	Description
Security Zone	
BC-7228	<p>Search results are now filtered according to the user's security zones as well as the search criteria and search-related permissions. In the search result, a user will see only entities that are in the same security zone as one of the user's groups unless the user has special permission to ignore security zones related to the entity being searched for.</p> <p>If an entity does not have a security zone specified (null security zone) then it will not be filtered according to security zone.</p> <p>Account, Producer, and Policy are entities that contain other entities such as Payment and Invoice. A search for a containing entity will be restricted by the security zone specified on that entity. A search for a contained entity will be restricted by the security zone specified on the containing entity.</p> <p>Entities that are contained by an account and are therefore restricted by an account security zone:</p> <ul style="list-style-type: none"> • DirectBillSuspenseItem • DelinquencyProcess • Disbursement <ul style="list-style-type: none"> • Account • Collateral • Invoice • Payment <ul style="list-style-type: none"> • DirectBillMoneyRcvd • SuspensePayment (if it has been applied to an account) • PaymentRequest <p>Entities that are contained by a producer and are therefore restricted by producer security zone:</p> <ul style="list-style-type: none"> • AgencyMoneyReceipt • Disbursement <ul style="list-style-type: none"> • Agency • Payment <ul style="list-style-type: none"> • AgencyBillMoneyRcvd • SuspensePayment (if it has been applied to a producer) • OutgoingProducerPayment <p>Entities that are contained by a policy and are therefore restricted by policy security zone:</p> <ul style="list-style-type: none"> • DelinquencyProcess <p>The following three new permissions enable a user to bypass security zone restrictions:</p> <ul style="list-style-type: none"> • ignoreacctsecurityzone • ignoreprodsecurityzone • ignorepolicysecurityzone <p>Example of searching for accounts:</p> <p>By default a user can find only accounts that the user created and that are in the same security zone as one of the user's groups. If the user has the grpacctsearch permission then they can also find accounts that were created by other users in their user group, provided those accounts are also in one of their security zones. If the user has the acctsearch permission then they can find all accounts regardless of which user or group created them, provided those accounts are in one of their security zones. If the user has the ignoreacctsecurityzone permission then their search results are not affected by security zones, although they are still affected by the acctsearch and grpaccountsearch permissions.</p> <p>Search security zone restrictions do not apply to the following entities:</p> <ul style="list-style-type: none"> • TroubleTicket • Activity • Contact • Transaction
Upgrade	
BC-8745	<p>Fixed a problem where duplicate item commissions (that is, item commissions that share the same invoice item, producer code, and role) were being consolidated during upgrade. However, the item commissions did not have the correct active status after consolidation. The active status is now set correctly during upgrade.</p>
BC-8746	<p>Fixed a problem with upgrade. Duplicate policy commissions were being consolidated, but the denormalized policy commission transfer fields (primary policy period, start date and end date) were not being set correctly after consolidation. These fields are now set correctly during upgrade.</p>

ID	Description
BC-8747	<p>Fixed a problem that could occur if a policy was been transferred multiple times. In BillingCenter 2.1.7 and earlier releases, BillingCenter may have created data which cannot be upgraded by our standard upgrade processing. If this is the case, there will be a message displayed at the start of upgrade indicating a data fix is required before proceeding. The message is: PolicyProducerCodeCharges found with erroneous Policy transfer transactions.</p>
	<p>Please contact Contact Guidewire Support for assistance with this problem before proceeding with upgrade.</p>
BC-8840	<p>Fixed a problem with the Commission Upgrade batch process that occurred when it was run with more than one worker.</p>
BC-8868	<p>In BillingCenter 3.0.5 and earlier releases, duplicate policy commissions (that is, policy commissions with the same policy period, role and producer code) were created when transferring the commission earner on a policy period back to a producer code that it had previously earned commission in the given role. Beginning with BillingCenter 3.0.6 we fixed this problem and changed upgrade to consolidate these duplicate policy commissions into a single policy commission of record. However, in certain cases some data would cause upgrade to fail. One of these cases occurred if there were <code>ItemEvents</code> where <code>FromPolicyCommissionID</code> or <code>ToPolicyCommissionID</code> pointed to a duplicate <code>PolicyCommission</code> that was about to be deleted. In that case we weren't correctly updating these foreign keys to the <code>PolicyCommission</code> of record. This left rows in the <code>ItemEvent</code> table that had foreign keys pointing to entities that had been deleted from the database, causing upgrade to fail.</p>
	<p>We now update these foreign keys to point to the policy commission of record and upgrade no longer fails.</p>
BC-8921	<p>Made the Commission Payable Calculations batch process thread safe.</p>
BC-8924	<p>If you are using SQLServer, you can use the new config.xml parameter, <code>MigrateToLargeIDsAndDatetime2</code>, to control whether you migrate to 64-bit BIGINT identifiers and datetime2 date columns.</p>
	<p>BillingCenter 7.0.1 uses 64-bit BIGINT identifiers and datetime2 date columns. BillingCenter versions prior to 7.0 use 32-bit signed INTEGER identifiers, which do not provide as many unique identifiers and datetime date columns. Converting all identifiers to 64-bit and all date columns to datetime2 is a time-intensive process, so by default BillingCenter 7.0.1 does not migrate to 64-bit identifiers. The new parameter controls whether this migration is done and is set to <code>false</code> by default.</p>
	<p>Guidewire recommends that you eventually complete the migration to 64-bit identifiers to ensure that there are enough unique identifiers available for BillingCenter. When you are ready to upgrade to using SQL Server 64-bit BIGINT identifiers and datetime2 date columns, see the <i>BillingCenter Upgrade Guide</i> topic <i>Enabling Migration to 64-bit IDs (SQL Server Only)</i> for instructions.</p>
BC-8960	<p>Fixed a problem where, during upgrade to BillingCenter 7.0.1, deprecated <code>CommissionWriteoff</code> transactions would be orphaned.</p>
	<p>Each <code>CommissionWriteoff</code> transaction is now converted during upgrade. These transactions will either be converted to a <code>CmsnPayableReduction</code> transaction or a <code>CmsnReceivableReduction</code> transaction, as appropriate.</p>
BC-9041	<p>When you upgrade to BillingCenter 7.0.1 from any earlier BillingCenter version, the work queue tables will be dropped.</p> <p>Work queue tables are used by the batch processes to store work items that need to be processed. A database upgrade should not be performed while any batch processes are running in BillingCenter.</p> <p>After the batch processes complete, failed work items will remain in the queue. Before upgrading, you have to run the workers again until all unfinished work items are completed, and hence removed from the work queue.</p>

ID	Description
BC-9146	<p>Changed upgrade so that we no longer create just one payment instrument for each account based on the currently selected payment method.</p> <p>If the payment method details for ACH or Credit Card have ever been filled in on the account, upgrade will now create one payment instrument (to represent both ACH and Credit Card). Upgrade will create this payment instrument regardless of the currently selected payment method.</p> <p>Because ACH and Wire both have the same payment method details, there is no way to determine whether the payment instrument should be set up for ACH or Wire, so we will always create a payment instrument for ACH.</p> <p>If the currently selected payment method is Misc, we set up a payment instrument for Misc. Otherwise, no Misc is created.</p>
Write-off	
BC-8952	Fixed a problem where, in BillingCenter 7.0.0, reversing a charge commission write-off would throw an exception and fail if the amount was higher than the user authorization limit. This error will no longer occur.

Platform Improvements and General Issues

Activities Assignment	
PL-18042	Removed user proximity assignment functions from the Assignable interface for BillingCenter as these functions rely on geocoded addresses and BillingCenter does not geocodes addresses in this release.
Application Integration	
PL-17945	Guidewire has changed the type of the URL field in the suite-config XSD definition file from xs:anyURI to xs:string.
Archiving	
PL-18992	As part of the changes made for file-based archiving, Guidewire now requires that you increment the version number stored in extensions.properties if you make changes to the data model and restart the application server. This change is optional in a development environment but mandatory in a production environment.
PL-17946	Guidewire now disallows duplicate entries in suite-config.xml. Guidewire defines duplicate entries as those with the same name, URL, and env values.
Assignment	
PL-13175	Delegated the UserRoleAssignment entity out to a UserRoleAssignmentDelegate and removed the UserRoleAssignment entity at the platform level. All Guidewire applications now own their own UserRoleAssignment entity and use entity delegation to UserRoleAssignmentDelegate.
PL-18196	Fixed assignGroupByRoundRobin to manage GroupAssignmentState with proper set of keys including intended GroupType to assign a group properly.
PL-18397	Fixed an issue that occurred during the creation of an exposure at the end of FNOL wizard if you were using assignGroupByRoundRobin assignment.
Batch Processes	
PL-13912	Guidewire has added the ability to set an env attribute on the <ProcessSchedule> element in scheduler-config.xml to specify the schedule to run in a certain environment. As a consequence, you can now have different results for batch processing based on environment. By default, Guidewire does not set the env attribute.
PL-15657	<p>Added Last Run Status column to the Processes table of the Batch Process Info page. This column allows a user to see if the last run of this batch task completed successfully or failed.</p> <p>The possible values are:</p> <ul style="list-style-type: none"> • Completed – The last run has completed successfully. • Not available – This task has not ran yet or running right now. • Failed/Interrupted –The last run has failed or was interrupted. (See the history of the process for more details.)
Clustering	

PL-10224	In addition to improvements to JGroups implementation of UNICAST protocol, this fix also introduced a new merge protocol, MERGE4. This is not set in the default protocol stack. (MERGEFAST is set.)
PL-17295	The valid range for multicast address is between 224.0.0.0 and 239.255.255.255. This change verifies the validity of the multicast address before attempting to join the cluster. If the address is invalid, BillingCenter throws the following exception: <pre>"Invalid multicast address " + multicastAddress + ". Valid range is between 224.0.0.0 and 239.255.255.255."</pre>
PL-17297	Guidewire has upgraded JGroups to the latest version, which is 2.12.1.
PL-17903	Guidewire now requires a unique server ID for each node in a cluster. The application verifies its server ID for uniqueness across all other cluster members as the node starts.
Code Utilities	
PL-18054	If you have used the \${x} substitution in document templates, then you need to revert those uses back to the <%= ... %> syntax.
Cognos Integration	
PL-18122	Modified the Guidewire reporting LDAP server to perform a case-insensitive compare of the user name in the Cognos LDAP authentication request to the users in BillingCenter database. (Guidewire normalizes user names to all lower case in its communications with Cognos.) The issue still exists that Cognos is not able to distinguish between Guidewire users that differ only in case even if these users are distinguished on Guidewire. Do not create Guidewire LDAP users that differ only in case.
Configuration Upgrade	
PL-18021	Fixed an issue with how the Upgrade tool automatically handled rule upgrades.
Database	
PL-18012	Previously, Guidewire prohibited you from connecting a development server to a production database, or connecting a production server to a development database. Guidewire now permits this kind of connection (merely generating a warning) if using the H2 database.
PL-18142	Fixed an issue in the Leap Year Days calculation that caused an infinite loop if times in the start and end of the DateRange did not match.
Database Upgrade	
PL-17732	Fixed performance issues with the upgrade process.
PL-17849	Guidewire has added a DateFromDatetime method to the BeforeUpgradeDBFunction utility class, which is available only in the context of upgrade queries. It is available only in restrictions, not in the query column list. Certain Guidewire applications need this for use in an upgrade trigger. In a BeforeUpgrade query, you can use this function in a SELECT statement, and it can take a nested function as an argument.
Entities/Metadata	
PL-10608	Guidewire now flags as an error in Studio—and as a warning at server start-up—if you attempt to set a non-queryable path on the Entity Path field in the Entity Names editor.
PL-17718	Fixed an issue in which using a forward slash (/) in a typecode definition prevented the application server from starting but Studio did not display an error flag for this condition.
PL-18071	Added support for denormalizing columns across tables to enhance the performance of search queries.
Geocoding/Proximity Search	
PL-17675	Fixed an issue in which the Bing Geocode plugin implementation was not visible in Studio.
PL-17886	Guidewire has deprecated the MapPoint implementation. Microsoft announced plans to retire the MapPoint web service effective November 18, 2011. If you currently use geocoding features and the MapPoint plugin, you must migrate from MapPoint to Bing Maps. Otherwise, geocoding features in the application will not function.
Gosu	
PL-16935	Fixed an issue in which BillingCenter threw a Null Pointer Exception if you attempted to change Gosu code in Studio and it was connected to the application server. (The issue was specific to functions in code blocks that were called from other PCF functions.)

PL-17478	Modified the Gosu Tester parser so that it treats scriptability modifiers in the same way that Gosu classes and enhancements do. This means you should no longer receive the following error message while running code in the Gosu Tester: 'The property [or method], "Xxxx", is not visible under the parser's visibility constraints.'
PL-17489	Fixed an issue in which certain corner cases of conditional variable declarations would fail with a Java byte-code VerifyError if the server was running in debug mode. This issue did not affect servers running in production mode.
PL-18033	Fixed an issue that caused incorrect behavior if you called reverse() on an implicitly-declared List in Gosu. (The issue did not occur with an explicitly declared List.) The issue was that the reverse sort on an implicitly-declared List occurred in-place, meaning that the sort occurred on the original list elements. The sort now occurs on a copy of the list, which is the desired behavior.
PL-18299	Added a preload mechanism to support pre-compilation of Gosu classes, as well as other primary classes in the system. The intent is to make the system more responsive the first time requests are made. To support this, Guidewire has added a Studio Other Resources → preload.txt file in which you can add a list of actions to take. The file contains static no-argument method calls, as well as the names of Gosu types to compile to byte-code or the Java types to load. Guidewire also added a new logging category of Server.Preload that provides DEBUG level logging of all actions during server pre-loading of Gosu classes.
GX Tools	
PL-12128	Added the ability for a web service to use types from a different WSDL. For example, <pre><xs:import namespace="http://example.com/gw/api/test/TheirAPI" schemaLocation="../../wsdl/local/gw/api/test/TheirAPI.wsdl" /></pre>
PL-18297	Restored missing eachException() methods on the Guidewire XML Modeler. Guidewire inadvertently removed these methods in a previous 7.x release.
History	
PL-17720	Guidewire no longer marks the HistoryType typelist as final. Therefore, it is now possible to add new values to this typelist. However, Guidewire made this change to allow the removal of the retrieved HistoryType in ClaimCenter. Guidewire recommends that you continue to add custom history types to the CustomHistoryType typelist, rather than the HistoryType typelist.
Logging	
PL-18234	Improved logging of ConcurrentDataChangeException. Now, the log message shows both of the following: <ul style="list-style-type: none">• The username of the current user, who received the ConcurrentDataChangeException• The previous user, who made the initial data modification The log writes the current user's name before the previous user's name in the message.
Messaging	
PL-18211	Fixed an issue in which messages could potentially be sent in incorrect send order if a bundle commit contained multiple event root entities.
Other - PL Services	
PL-14011	The (Server Tools) Management Beans → CurrentUserSessions information now displays a sorted list of users. If a user is logged in more than once, BillingCenter appends the number of sessions to the user name in parentheses. For example: psmith(2),bsmith The intent is to preserves all information that is currently available, while providing a view that is easier to scan because of the sorting and the non-repetition of names.
Persistence	
PL-17656	Guidewire has introduced a new class com.guidewire.external.Type to use for Java API methods that require an instance of com.guidewire.external.Type. You can obtain an instance of this class using the com.guidewire.external.Type#of method by passing in the class representing the desired type. Methods that require an instance of com.guidewire.external.Type need to document what kinds of types are expected.
Plugins	

PL-17162	Guidewire has added interface <code>gw.plugin.SharedBundlePlugin</code> . Any plugin that implements this interface shares the current bundle rather than creating a new one. For example, any plugin that implements <code>IPreUpdateHandler</code> now needs to add <code>gw.plugin.SharedBundlePlugin</code> to its list of implemented interfaces.
Queries	
PL-17676	Guidewire has deprecated the <code>RawQuery</code> property on <code>gw.api.database.Query</code> . Instead, create a <code>gw.api.filters.StandardQueryFilter</code> and use its <code>filterQuery</code> method or the <code>IQueryBeanResult.addFilter</code> method to apply the filter to a query.
PL-17719	
Rules Infrastructure	
PL-17893	Guidewire has added a new (Server Tools) Info Page that shows the safe persisting order of the Preupdate rules. This page lists the order in which BillingCenter runs the Preupdate rules for their root entities.
Security	
PL-10126	<p>Guidewire has added a new Credentials plugin to the base configuration. The <code>CredentialsPlugin</code> is similar to the <code>DBAuthenticationPlugin</code> and consists of the following three components:</p> <ul style="list-style-type: none"> • Interface <code>CredentialsPlugin</code> with the following sole method: <code>public UsernamePasswordPairBase retrieveUsernameAndPassword(String key)</code> • Class <code>CredentialsUtil</code> with the following sole static method: <code>public static UsernamePasswordPairBase getCredentialsFromPlugin(String key)</code> • Class <code>CredentialsPlugin.gs</code> provides a default implementation of the plugin in the base configuration. <p>Code can call the method <code>CredentialsUtil.getCredentialsFromPlugin</code>, passing in a key to specify for which application the credentials are sought. The static method returns the username/password pair for the desired key. The default implementation currently reads the username/password pair from a <code>Credentials.xml</code> file:</p> <ul style="list-style-type: none"> • If the code encounters the token <code> \${username} </code> as the value of the username value in the properties file or the token <code> \${password} </code> as the value of the password value in the properties file, then the code uses the <code>Credentials</code> plugin to obtain the real username and password. • If the <code>Credentials</code> plugin is not enabled or these tokens are not present, the code simply uses the values that are present in the properties file (for backwards compatibility). <p>However, you can substitute any mechanism for retrieving username/password pairs in place of the default mechanism. This can include retrieving username/password pairs from a directory service or from encrypted fields in a database.</p> <ul style="list-style-type: none"> •
Studio IDE - Debugger	
PL-17929	Fixed an issue that caused a Null Pointer Exception that caused the interface to freeze under some circumstances after connecting Studio to the application server or starting a debugging session on an already-running server.
PL-18106	<p>Fixed an issue in which some actions in Studio, such as setting breakpoints, would occasionally block for a short while waiting for a server response, but did not provide any visible feedback.</p> <p>Guidewire now displays a progress dialog to indicate the delay.</p>
PL-18110	<p>Previously, if the connection between Studio and the application server was lost (for example, if the server session was terminated for some reason), Studio did not terminate the debug session. As a consequence, after server restart, it was possible for a user to believe that debugging was still ongoing, even though, in reality, the debugger needed to be reset.</p> <p>The new behavior is that Studio properly terminates the debugger session if it loses connection to the server and notifies the user. The user needs to manually reactivate the debugger after connection to the application server is restored.</p>
PL-18112	<p>Previously, if there was a web service error while activating the Server debugger, it was possible for the activation to fail, but for Studio to show the debugger as active, even it was not. Studio now notifies you of the failure and properly puts the debugger into an inactive state.</p> <p>Guidewire has also put into place additional recovery mechanisms from certain types of web services errors, making it less likely for the debugger connection to fail in the first place.</p>
PL-18116	Fixed an issue that caused Studio to hang if you were debugging the server and the Gosu Tester debugger was also active. It also occurred if you have an active Gosu Tester debug session, then opened the Guidewire Studio connection dialog and immediately canceled. In that case Studio terminated the Gosu Tester debug session as well.

PL-18275	Fixed a problem with the remote tester debugger hanging if you disconnected it from the server during an active debug session.
PL-18298	Fixed a problem in which it was possible for the Studio tester window to freeze up if the user was using the tester debugger while connected to a remote server and modified some code in Studio and saved it.
Studio IDE - Other	
PL-16798	Modified the Rule Condition parser to accept a statement list, instead of a simple expression only. However, the statement list must contain a return statement. For example:
	uses java.util.HashSet uses gw.lang.reflect.IType
	var o = new HashSet<IType>() {A, B, C, ...} return o.contains(typeof(...))
PL-17386	Added the ability to inspect inner class values in Gosu while debugging in Studio.
PL-17606	Fixed an issue in which Studio incorrectly handled creating a View entity extension.
PL-18107	Studio now shows a broken server connection state (icon) if it loses a server connection, for example, if the user terminates the server process. <ul style="list-style-type: none"> It detects lost connections very quickly if debugging as there is immediate feedback from the debug API. It detects lost connections more slowly (but, with not more than 1 minute of delay), if not debugging, as Studio queries the server for status at one minute intervals. If Studio detects the server has restarted, Studio reconnects automatically.
PL-18109	Previously, if Studio was connected to a server that was not responding for a period of time, multiple threads would pile up in the Studio process waiting for the server to respond. Guidewire has improved its handling of restoring Studio's server connection to a server that was previously connected, but became unavailable for a period of time, and then became available again.
PL-18198	Fixed an issue in which, if you set breakpoints in a Studio session, Studio forgot the breakpoints between sessions if you activated the Server debugger after starting the application server, and then shut down and restarted Studio.
PL-18227	Fixed an issue in which BillingCenter stored the initial value of a newly created script parameter in an incorrect location, thus invalidating the application installation checksum. BillingCenter correctly stored subsequent entries in the correct location.
Studio IDE - PCF Editor	
PL-11489	Guidewire has added a new PCF Verification option (Tools → Options → Verification Options) that you use to enable/disable whether BillingCenter limits the number of included sections in second-pass PCF verification. (Second-pass verification can cause performance issues with combinations of modal PCF files.) For details, see "Setting Verification Options" in the <i>BillingCenter Configuration Guide</i> .
PL-17576	Fixed an issue that caused Studio to throw an <code>AssertionError</code> under certain circumstances. This generally occurred if you attempted to load a file (a PCF file, for example) modified outside of Studio into Studio with a stale version of that file already open in Studio. Studio would throw this error upon your return to Studio. This could happen: <ul style="list-style-type: none"> If you opened a file in Studio, shifted focus to another application such as IntelliJ and modified and saved the file, then shifted focus back to Studio. If you opened a file in Studio and shifted focus away from Studio, then (through a source control management system), reloaded a modified version of the file into the local file system and shifted focus back to Studio. In each case, there was a mismatch between the file stored locally and the version of the file contained in Studio's memory as Studio did not save the open file properly on losing focus.
Utilities	
PL-17920	Guidewire has removed the following JAR files from the application build: <ul style="list-style-type: none"> activation.jar activation-1.1.1.jar
PL-17922	Guidewire has removed the following JAR files from the Guidewire application builds because the standard Java EE install contains these files: <ul style="list-style-type: none"> mailapi.jar smtp.jar
Web - Other	

PL-17861	Added a disablePostOnEnter attribute to the Input Group widget.
Web - UI/Runtime	
PL-18233	Fixed an issue that failed to highlight in yellow a problematic RadioButtonCell field even if the data value for the RadioButtonCell contained an error.
Web Services - WSI	
PL-12215	When you place a WSDL from a web service publisher anywhere in the Gosu class path, Guidewire generates Gosu types from the WSDL elements. A problem in which these generated Gosu types could not be used successfully has been fixed.
PL-16577	To authenticate using HTTP authentication on WebLogic, add the following inside the security-configuration tag of WebLogic's config.xml: <enforce-valid-basic-auth-credentials>false</enforce-valid-basic-auth-credentials>
PL-17601	Restored the functionality to extend an interface by a Gosu type. The WSDL was incorrectly overriding the final methods of the Object class (wait, notify, and similar methods). Eventually, when these WSDLs were transformed into Java classes, the compilation of the Java class failed because the final methods on Object could not be overridden. The fix was to not override these methods in WSDL generation.
PL-17902	Guidewire has added a check for the existence of the generated WSDL directories.
Workflow	
PL-9668	<p>Added new configuration parameter WorkflowLogDebug, which takes a Boolean value:</p> <ul style="list-style-type: none"> If set to true, BillingCenter outputs the ordinary verbose system workflow log messages from the Guidewire server to the workflow log. If set to false, BillingCenter does not output any of the ordinary system messages. <p>The setting of this parameter does not have any effect on calls to log workflow messages made by customers. Therefore, all customer log messages are output. If customers are experiencing too many workflow messages being written to the bc_workflowlog table, Guidewire recommends that you set this parameter to false.</p>
PL-17556	<p>It is now possible to run the following writer batch processes from either the (Server Tools) Batch Process Info or (Server Tools) Work Queue Info pages:</p> <ul style="list-style-type: none"> Activity Escalation Group Exception User Exception
XML Element (and XSD types)	
PL-17934	Fixed an issue in which the XML subsystem did not parse complex types correctly if the complex types within XSD files made circular references to each other.

Known Issues and Limitations

This section describes known issues with this release of Guidewire BillingCenter.

- BillingCenter Known Issues
- Platform Known Issues

BillingCenter Known Issues

Commission: Transferring an agency bill item between producers and then to a direct bill payer account affects pay-on-pay commission earning (BC-218)

Issue – Transferring an agency bill invoice item back to original producer code (after a round-trip of transfers), and then switching the item to direct bill does not reearn pay-on-pay commission on that producer code.

Workaround – Guidewire is aware of this issue.

Performance: Large amount of data causes some screens to render slowly or not at all (BC-720, BC-765)

Issue – Large amounts of data cause the new payment screens (New Direct Bill Payment, New Direct Bill Credit Distribution) and the Invoices screen to render slowly and throw an error.

Workaround – Guidewire is aware of this issue.

Integration: PolicyPeriod.getStatus() to come from an integration point (BC-1582)

Issue – When a policy period is cancelled, BillingCenter shows the policy period as In-Force on the Account and Policy screen, which is misleading.

Workaround – Get the status from the Policy Administration System.

Integration: Null constraint violations when adding and removing account and producer contact using the UI (BC-1788)

Issue – When a policy period is cancelled, BillingCenter shows the policy period as In-Force on the Account and Policy screen, which is misleading.

Workaround – Add the following Gosu to AccountDetailsContacts.pcf in the code section:

```
function addNewContact(accountContact : AccountContact) {
    if(accountContact.Contact == null) {
        var contact = new BCContact()
        accountContact.Contact = contact
        accountContact.Contact.Type = ContactType.TC_PERSON
    }
    account.addToContacts(accountContact)
}
In ProducerContacts.pcf in the code section:
function addNewContact(producerContact : ProducerContact) {
    if(producerContact.Contact == null) {
        var contact = new BCContact()
        producerContact.Contact = contact
        producerContact.Contact.Type = ContactType.TC_PERSON // <---- add this line
    }
    producer.addToContacts(producerContact)
}
```

Payment: Calling reverse on the DirectBillMoneyRcvd does not reverse an associated distribution (BC-1885)

Issue – DirectBillMoneyRcvd has two methods that reverse it - `reverse` and `reverseMoneyAndDistribution`. Calling `reverse` alone does not reverse the distribution associated with the money if there is one, so calling `reverse` can leave the data in a bad state.

Workaround – To reverse a DirectBillMoneyRcvd, do not use the `reverse` method. Use `reverseMoneyAndDistribution` method instead.

Billing Instruction: BillingCenter should require a term number (BC-3115)

Issue – Currently if a BillingInstruction submits a policy period for creation and the policy period's term number is null, BillingCenter attempts to use the value “1” as the term number. This can lead to errors because that term number may already exist.

Workaround – Always specify the term number for the policy period if you are sending a billing instruction from the policy administration system. Then BillingCenter will not attempt to use the default value “1”.

If BillingCenter enforces this requirement in the future, you will not have to change your integration code.

User Interface: Error selecting charge-Level override after moving back and forward through wizard (BC-3697)

Issue – Error selecting charge-level override after moving back and then forward again through wizard while creating a policy charge. This occurs for both the new policy wizard and the policy change wizard.

Workaround – Fill in the first screen completely before moving to the next. Avoid the user interface and use integration API instead.

ContactManager Integration: Extra messages after contact update (BC-3867)

Issue – After ContactManager calls `updateContact`, BillingCenter makes a redundant update call back to ContactManager.

Workaround – No action required. Ignore extra messages.

ContactManager Integration: Add Existing Contact in New Policy Wizard does not allow adding roles to the contact. (BC-8716)

Issue – When a new policy is being created and an existing contact is selected to be added to the policy, then the user is not given an opportunity to add roles to that contact.

Workaround – Finish creating the policy (with no roles on the added contacts). View the policy, click **Contacts**, click **Edit**, select the added contact, and add the desired roles.

ContactManager Integration: Contact merge can sometimes leave the account with no primary contact. (BC-8717)

Issue – If ContactManager sends a request to BillingCenter to merge contacts, it passes a surviving contact and dissolving contact. BillingCenter keeps the surviving contact and retires the dissolved one. If the dissolving contact happens to be the primary contact for an account, then surviving contact should be made the primary contact for an account. Instead, the merge ends without making this change. As a result, after the merge, there are no primary contacts.

Workaround – Edit the surviving contact on the account and make it primary.

Agency Bill: Commission should not become payable, agency bill with assignment of items and commission override. (BC-9039)

Issue – If the following events happen in this order on an agency bill policy period, the rate change will not happen correctly:

1. Point in time transfer of an agency bill policy between two producers
2. Commission rate change on a charge that was transferred

The total reserves and expenses will be set properly, but a portion of the reserve will be earned at the time the rate change occurs, even if it should be earned at a later date.

Workaround – Contact Guidewire Support for assistance.

Agency Bill: Agency bill payment search does not work with the returned policy period. (BC-9107)

Issue – The **Agency Bill Payment Wizard** has a policy period search option to be used when adding items. When you select a policy period and then click search to find invoice items, none are found.

Workaround – Use the policy without the period to search for invoice items. For example, if you want to find invoice items for **AutoPolicy-1**, remove the policy period number, and search for **AutoPolicy** instead.

Write-off: Charge reversal with a write-off is adding the write-off amount to paid from account transaction. (BC-9114)

Issue – Reversing a charge with a write-off is adding the write-off amount to the PaidFromAccount transaction.

Workaround – Manually reverse the write-off prior to reversing the charge.

Agency Bill: The balance on agency bill cycles has incorrect amount after upgrade if gross/commission written off. (BC-9175)

Issue – The balance on agency bill cycles is incorrect after upgrade when part of the statement amount has been written off (either the gross or the commission).

Workaround – Guidewire is aware of this issue.

Charge hold on a delinquent policy reverses the past due charge and not the delinquency. (BC-9198)

Issue – Creating a charge hold on a delinquent policy reverses the past due charge and not the delinquency. It is creating offset and onset invoice items, but the invoice item on the carried forward Invoice is not getting any billed or due transactions. This leaves the account due at a positive balance and the delinquency remains.

Workaround – Use a TroubleTicket hold. Note that the trouble ticket hold will not cause the Delinquency to stop.

Agency Bill: Total of all item commission reserves are used when system should use only reserves from one producer code. (BC-9246)

Issue – If the following two events happen in this order on a policy period, then the commission reserve is being distributed improperly across items.

1. Point in time transfer of an agency bill policy between two producers
2. An item gross amount (items from the policy) is edited

The total reserves and expenses at the charge level are correct, but the distribution across invoice items will be incorrect.

Workaround – Contact Guidewire Support for assistance.

Platform Known Issues

First time you click the arrow of the typekey input, the drop-down menu does not open (PL-10134)

Issue – The drop-down menu does not open on the first click of the arrow on a typekey input. Instead, the help text opens.

Workaround – Turn off help text on focus by setting InputHelpTextOnFocus to false in config.xml. After you do that, the help text shows only if you mouse over the input, and it does not interfere with opening a drop-down menu.

XML API upgrade feature missing from documentation (PL-10257)

Issue – The *Integration Guide* describes a new set of XML APIs based on the XmlElement class. (Legacy APIs are based on the XmlNode class.) You can continue to use the legacy APIs. However, the *Integration Guide* omits mentioning an additional upgrade-specific feature.

Workaround – For backwards compatibility only, you can import an XML schema into the Gosu type system using the legacy XML system by following these instructions:

1. Copy:

BillingCenter/modules/bc/config/registry/compatibility-xsd.xml

To:

`BillingCenter/modules/configuration/config/registry/compatibility-xsd.xml`

2. Add an entry for your schema. Set the value of the `namespace` attribute to the Gosu package name of the schema. For example, if the schema is in the package location `my.package` and is called `myschema.xsd`, set the value of `namespace` to `my.package.myschema`.

[Studio rules do not use correct capitalization for root object's name \(PL-10740\)](#)

Issue – Rule set root objects are not named with first letter lower-cased.

Workaround – The Rules engine issues a warning if the correct case for objects is not being used.

[Countries configured in zone-config.xml still generate a warning during regen-dictionary even when zone data is loaded for all of these countries \(PL-11947\)](#)

Issue – Countries configured in `zone-config.xml` still generate a warning during regen-dictionary even when zone data is loaded for all of these countries.

Workaround – Warning message is created in error and can safely be ignored.

[User interface cannot handle starting multiple instances of a batch process \(PL-12372\)](#)

Issue – The user interface cannot handle starting multiple instances of a batch process.

Workaround – To execute multiple instances of a batch process, start them from the command line. Also, to allow multiple instances to be run simultaneously, you must ensure that the `BatchProcess.isExclusive()` method returns `false`.

[Type system refresh after PCF page title change does not update corresponding menu label \(PL-13057\)](#)

Issue – The type system refresh after a PCF page title change does not update the corresponding menu label.

Workaround – After updating a page title, restart the server to refresh menu labels and avoid null pointer exceptions due to stale references.

[Length limitation on entity localization table names \(PL-13360\)](#)

Issue – There is a length limitation on entity localization table names.

Workaround – Ensure that the localization `tableName` property specified in the entity extension file is less than 16 characters. If the localization table name exceeds the maximum length, the error message indicates that 18 characters are allowed. However, the error message does not account for two additional characters added by the application.

[US-Locations.txt file with the US geodata from GreatData has special characters that cause validation problems with United States Postal Service \(USPS\) data \(PL-13384\)](#)

Issue – The `US-Locations.txt` file contains information that does not conform to United States Postal Service (USPS) standards for bulk mailings.

Workaround – The provided `US-Locations.txt` file is intended only for use in geocoding to identify addresses for a location. You can edit the `US-Locations.txt` file to conform to your particular address standards, and then import that version of the file instead.

GX models that reference virtual fields and enhancements throw null pointers if null (PL-13560)

Issue – The GX models that reference virtual fields and enhancements throw null pointers when these fields and enhancements are null.

Workaround – Include null checks and error handling to prevent referenced virtual fields or enhancements that are null from causing null pointer exceptions.

Sending email with file attachment with unicode filename is not correctly handed over to the mail server (PL-13582)

Issue – An email with a file attachment that has a unicode file name is not sent to the mail server correctly.

Workaround – Use Latin characters for file names on attached files.

GX model generated XSD cannot be parsed by JAXB (PL-13598)

Issue – XSD generated by the GX model cannot be parsed by JAXB.

Workaround – Add JAXB annotation elements to the XSD to specify the necessary metadata, such as class names, to enable JAXB to generate the Java class files. Contact Guidewire Support for a sample XSD file that is annotated this way.

Cannot make a field from a delegate into a localized column (PL-13761)

Issue – You cannot make a field from a delegate into a localized column.

Workaround – Move the column to be localized off the delegate and onto each of the implementing entities. Then, to make the column appear as though it exists on the delegate, define an enhancement property on the delegate that *delegates* to the appropriate column, depending on the implementing entity.

Studio test functionality not working correctly (PL-15153)

Issue – If you run a test in the Tests folder in the Studio Resources tree, you get the following exception:

```
Using Test Environment Delegate: com.guidewire.testharness.ConfigEnvTestEnvironmentDelegate
gw.internal.gosu.parser.RuntimeExceptionWithNoStackTrace: java.lang.ClassNotFoundException:
    qa.DothisTest
Caused by: java.lang.ClassNotFoundException: qa.DothisTest
at gw.internal.gosu.parser.TypeLoaderAccess.getIntrinsicTypeByFullName(TypeLoaderAccess.java:522)
at gw.internal.gosu.parser.TypeSystemImpl.getByFullName(TypeSystemImpl.java:139)
at gw.lang.reflect.TypeSystem.getByName(TypeSystem.java:116)
at com.guidewire.studio.junit.RunTestCommand.buildTestSuite(RunTestCommand.java:109
...

```

Workaround – Run the following command before you run a test:

```
.../bin gwbc dev-deploy
```

Problem with *regen-java-api* command and JAR files (PL-16351)

Issue – If you run the *BillingCenter/bin/gwpc regen-java-api* command, BillingCenter creates a *BillingCenter/java-api/lib* directory and puts JAR files for Java development in this directory. However, BillingCenter does not always perform this task correctly, resulting in some of the generated JAR files' not being copied to this directory.

Workaround – For missing library files, copy or import into your Java development environment the missing library files from the following directory:

```
BillingCenter/admin/lib
```

Copy them into the following directory:

```
BillingCenter/java-api/lib
```

Renaming method or property throws ParseResultsException (PL-16633)

Issue – If you rename a property or a method or you change a method signature, and a workflow references that property or method in a Gosu field, BillingCenter throws `ParseResultsException`. This is the intended behavior.

Workaround – Restart the workflow engine. To do so:

1. Log into BillingCenter using an administrative account.
2. Access Internal Tools → Reload.
3. Click Reload Workflow Engine.

Deploying EAR File on WebSphere 7.0.0.15 Generates Error Message (PL-18613)

Issue – The following steps generate an error message if you attempt to deploy a Guidewire-generated EAR file to WebSphere 7.0.0.15.

1. Generate an EAR file using the following command:

```
gwbc build-websphere-ear
```

2. Deploy the EAR file to WebSphere.

The deployment fails with an error message similar to the following:

```
com.ibm.websphere.management.application.client.AppDeploy  
Exception: com.ibm.websphere.management.application.client.AppDeploymentException:  
ADMA0207E: EE 5 module ab.war in ear file contains unsupported xmi format bindings file.
```

Workaround – Add the following at the top of the `web.xml` file before attempting to generate the EAR file.

```
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"  
"http://java.sun.com/dtd/web-app_2_3.dtd">
```

Localization Issues

Different date separators in localization cause dates to function differently (PL-18230)

Issue – Changing the short `DateFormat` in `localization.xml` from the default `MM/dd/yyyy` to `dd.MM.yyyy` can cause the following issue. If you choose a date field in BillingCenter with a pre-filled date, then the date picker disregards the pre-filled date and uses the current date instead. This issue also occurs if you change the short `DateFormat` to `MM-dd-yyyy` as well.

This is not an issue if you are using the ICU libraries for locale formatting.

Workaround – In the display key that defines valid date characters, replace the forward slash with a period:

1. Open `display.properties` for editing and find the following display key:

```
Javascript.Calendar.ValidDateCharacters
```

2. Change its value from:

```
Javascript.Calendar.ValidDateCharacters=0123456789 \:/AaPpMm
```

to:

```
Javascript.Calendar.ValidDateCharacters=0123456789 \:.AaPpMm
```




chapter 28

Guidewire BillingCenter 7.0.0 Release Notes

Release 7.0.0

Document Revision: 02-July-2015

Copyright © 2001-2015 Guidewire Software, Inc. All rights reserved.

Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire Reinsurance Management, Guidewire ContactManager, Guidewire Vendor Data Management, Guidewire Client Data Management, Guidewire Rating Management, Guidewire InsuranceSuite, Guidewire ContactCenter, Guidewire Studio, Guidewire Product Designer, Guidewire Live, Guidewire DataHub, Guidewire InfoCenter, Guidewire Standard Reporting, Guidewire ExampleCenter, Guidewire Account Manager Portal, Guidewire Claim Portal, Guidewire Policyholder Portal, Gosu, Deliver Insurance Your Way, and the Guidewire logo are trademarks, service marks, or registered trademarks of Guidewire Software, Inc. in the United States and/or other countries.

All other trademarks are the property of their respective owners.

This material is confidential and proprietary to Guidewire and subject to the confidentiality terms in the applicable license agreement and/or separate nondisclosure agreement.

Guidewire products are protected by one or more United States patents.

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire BillingCenter.

Version Number

This release of Guidewire BillingCenter is 7.0.0.

- For Standard Reporting, BillingCenter 7.0 supports IBM Cognos Business Intelligence 10.1 as the report server.
- If you are integrating with Guidewire PolicyCenter, this version of BillingCenter requires PolicyCenter 7.0.0 or later. If you wish to integrate with PolicyCenter 7.0.0 or 7.0.1, see “Cannot integrate with PolicyCenter 7.0.0 or 7.0.1 (BC-8554)”.

Installing This Release

For installation information, see the *BillingCenter Installation Guide*.

Changes in this Release

This section describes the product changes in this release.

- Base PCF File Changes
- Changes in This Release Provided in Upgrade Diff Report

Base PCF File Changes

All links below require the `readme_files` directory on your local disk.

BillingCenter release 3.0.6 to 7.0.0

- To view a report of the changes in the base PCF files in the `modules/bc` directory, [click here](#).
- To view a report of the changes in the base PCF files in the `modules/p1` directory, [click here](#).

Changes in This Release Provided in Upgrade Diff Report

Guidewire provides a report detailing certain differences between the current release and your prior release. This report describes changes in display keys, entities, typelists, and the Gosu API. To obtain your custom Upgrade Diff Report, visit the Guidewire Resource Portal.

Known Issues and Limitations

This section describes known issues with this release of Guidewire BillingCenter.

- BillingCenter Known Issues
- Platform Known Issues

BillingCenter Known Issues

Commission: Transferring an agency bill item between producers and then to a direct bill payer account affects pay-on-pay commission earning (BC-218)

Issue – Transferring an agency bill invoice item back to original producer code (after a round-trip of transfers), and then switching the item to direct bill does not reearn pay-on-pay commission on that producer code.

Workaround – Guidewire is aware of this issue.

Performance: Large amount of data causes some screens to render slowly or not at all (BC-720, BC-765)

Issue – Large amounts of data cause the new payment screens (New Direct Bill Payment, New Direct Bill Credit Distribution) and the Invoices screen to render slowly and throw an error.

Workaround – Guidewire is aware of this issue.

Payment: Calling reverse on the DirectBillMoneyRcvd does not reverse an associated distribution (BC-1885)

Issue – DirectBillMoneyRcvd has two methods that reverse it - reverse and reverseMoneyAndDistribution. Calling reverse alone does not reverse the distribution associated with the money if there is one, so calling reverse can leave the data in a bad state.

Workaround – To reverse a DirectBillMoneyRcvd, do not use the reverse method. Use reverseMoneyAndDistribution method instead.

ContactManager Integration: Trouble ticket search error (BC-2246)

Issue – Trouble ticket search throws an exception if the user searches using address, city, country, and zip.

Workaround – Guidewire is aware of this issue.

Billing Instruction: BillingCenter should require a term number (BC-3115)

Issue – Currently if a BillingInstruction submits a policy period for creation and the policy period's term number is null, BillingCenter attempts to use the value “1” as the term number. This can lead to errors because that term number may already exist.

Workaround – Always specify the term number for the policy period if you are sending a billing instruction from the policy administration system. Then BillingCenter will not attempt to use the default value “1”.

If BillingCenter enforces this requirement in the future, you will not have to change your integration code.

User Interface: Error selecting charge-Level override after moving back and forward through wizard (BC-3697)

Issue – Error selecting charge-level override after moving back and then forward again through wizard while creating a policy charge. This occurs for both the new policy wizard and the policy change wizard.

Workaround – Fill in the first screen completely before moving to the next. Avoid the user interface and use integration API instead.

ContactManager Integration: Extra messages after contact update (BC-3867)

Issue – After ContactManager calls updateContact, BillingCenter makes a redundant update call back to ContactManager.

Workaround – No action required. Ignore extra messages.

Commission: After transferring a policy between producers and back again, some commission reserve not automatically reearned (BC-4458)

Issue – After a policy is transferred back and forth between two producers, some of the commission reserve may not automatically reearn.

Workaround – Guidewire is aware of this issue.

Cannot integrate with PolicyCenter 7.0.0 or 7.0.1 (BC-8554)

Issue – Integration with PolicyCenter 7.0.0 or PolicyCenter 7.0.1 requires configuration changes.

Workaround – Please contact Guidewire Customer Support for assistance.

ContactManager Integration: Contact merge can sometimes leave the account with no primary contact. (BC-8717)

Issue – If ContactManager sends a request to BillingCenter to merge contacts, it passes a surviving contact and dissolving contact. BillingCenter keeps the surviving contact and retires the dissolved one. If the dissolving contact happens to be the primary contact for an account, then surviving contact should be made the primary contact for an account. Instead, the merge ends without making this change. As a result, after the merge, there are no primary contacts.

Workaround – Edit the surviving contact on the account and make it primary.

Platform Known Issues

Issues with Internet Explorer 9

Issue – If you are using the Internet Explorer 9 browser, it is possible to see issues such as screen flickering or an incorrect tab order for fields. According to public reports, IE 9 exhibits these and other issues with a variety of web sites and web applications.

Workaround – Because this is the behavior of the Internet Explorer 9 rendering engine, Guidewire cannot address these issues. However, there are reports of IE 9 users being able to reduce these issues by changing the new IE 9 Accelerated Graphics settings on the Advanced tab of the Internet Options dialog.

First time you click on the arrow of the typekey input, the drop-down menu will not open (PL-10134)

Issue – The drop-down menu does not open on the first click of the arrow on a typekey input. Instead, the help text opens.

Workaround – Turn off help text on focus by setting InputHelpTextOnFocus to false in the config.xml file. By doing that, the help text shows only if you mouse over the input and will not interfere with opening a drop down menu.

XML API upgrade feature missing from documentation (PL-10257)

Issue – The *Integration Guide* describes a new set of XML APIs based on the XmlElement class. (Legacy APIs are based on the XmlNode class.) You can continue to use the legacy APIs. However, the *Integration Guide* omits mentioning an additional upgrade-specific feature.

Workaround – For backwards compatibility only, you can import an XML schema into the Gosu type system using the legacy XML system by following these instructions:

1. Copy:

BillingCenter/modules/pl/config/registry/compatibility-xsd.xml

To:

`BillingCenter/modules/configuration/config/registry/compatibility-xsd.xml`

2. Add an entry for your schema. Set the value of the `namespace` attribute to the Gosu package name of the schema. For example, if the schema is in the package location `my.package` and is called `myschema.xsd`, set the value of `namespace` to `my.package.myschema`.

['ListDetailPanel' throws exception \(PL-10316\)](#)

Issue – It is possible for BillingCenter to throw an exception if the user cancels out of a `ListDetailPanel` widget if `StartEditMode` is also True.

Workaround – Set `StartEditMode` to False for the screen that contains the `ListDetailPanel`. As a consequence, the user must click `Edit` to modify that screen.

[Countries configured in 'zone-config.xml' still generate a warning during 'regen-dictionary' even when zone data is loaded for all of these countries \(PL-11947\)](#)

Issue – Countries configured in `zone-config.xml` still generate a warning during regen-dictionary even when zone data is loaded for all of these countries.

Workaround – Warning message is created in error and can safely be ignored.

[There is a length limitation on entity localization table names \(PL-13360\)](#)

Issue – There is a length limitation on entity localization table names.

Workaround – Ensure that localization `tableName` property specified in the entity extension file is less than 16 characters. The error message generated if the localization table name exceeds the maximum length indicates that 18 characters are allowed, but that does not account for two additional characters added by the application.

[GX models that reference virtual fields and enhancements throw null pointers if null \(PL-13560\)](#)

Issue – The GX models that reference virtual fields and enhancements throw null pointers when they are null.

Workaround – Ensure that null checks and error handling is included so that if referenced virtual fields or enhancements are null, then there will not be a null pointer exception.

[Sending email with file attachment with unicode filename is not correctly handed over to the mail server \(PL-13582\)](#)

Issue – Sending email with file attachment with unicode filename is not sent to the mail server correctly.

Workaround – Use Latin characters for file names on attached files.

['JavaToolkit.gs' has incorrectly hard coded memory, which results in failed 'regen-java-api' Ant task \(PL-13663\)](#)

Issue – The `JavaToolkit.gs` has hard coded memory which can result in failed `regen-java-api` Ant tasks.

Workaround – Increase the size of the maximum heap setting on line 161 of `JavaToolkit.gs` in the Ant module. The default value is 512.

[Problem with 'regen-java-api' command and JAR files \(PL-16351\)](#)

Issue – If you run the `BillingCenter/bin/gwpc regen-java-api` command, BillingCenter creates a `BillingCenter/java-api/lib` directory and puts JAR files for Java development in this directory. However, BillingCenter does not always perform this task correctly, resulting in some of the resulting JAR files not being copied to this directory.

Workaround – For missing library files, copy (or import into your Java development environment) the missing library files from the following directory:

`BillingCenter/admin/lib`

into the following directory:

`BillingCenter/java-api/lib`

[Database Takes Extremely Long Time to Upgrade \(PL-17732\)](#)

Issue – Guidewire has modified the way that it calculates denormalized columns. As a consequence, you can experience performance issues with the upgrade process under either of the following circumstances:

- You are using SQL Server as the database.
- You are using Oracle as the database. But, you are not using a Japanese locale, nor are you implementing accent-insensitive searching.

Workaround – Guidewire is aware of this issue.



chapter 29

Guidewire BillingCenter 3.0.7 Release Notes

Release 3.0.7

Document Revision: 02-July-2015

Copyright © 2001-2015 Guidewire Software, Inc. All rights reserved.

Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire Reinsurance Management, Guidewire ContactManager, Guidewire Vendor Data Management, Guidewire Client Data Management, Guidewire Rating Management, Guidewire InsuranceSuite, Guidewire ContactCenter, Guidewire Studio, Guidewire Product Designer, Guidewire Live, Guidewire DataHub, Guidewire InfoCenter, Guidewire Standard Reporting, Guidewire ExampleCenter, Guidewire Account Manager Portal, Guidewire Claim Portal, Guidewire Policyholder Portal, Gosu, Deliver Insurance Your Way, and the Guidewire logo are trademarks, service marks, or registered trademarks of Guidewire Software, Inc. in the United States and/or other countries.

All other trademarks are the property of their respective owners.

This material is confidential and proprietary to Guidewire and subject to the confidentiality terms in the applicable license agreement and/or separate nondisclosure agreement.

Guidewire products are protected by one or more United States patents.

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire BillingCenter.

Version Number

This release of Guidewire BillingCenter is 3.0.7

For Standard Reporting, this release supports InetSoft StyleReport Enterprise Edition 10.1, 20100113.

Support

For assistance, visit the Guidewire Resource Portal – <http://guidewire.custhelp.com>

Installing This Release

For installation information, see the *BillingCenter Installation Guide*.

Changes in this Release

This section describes the product changes in this release.

- Base PCF File Changes
- Improvements to Upgrade Diff Report
- BillingCenter Improvements and General Issues
- Platform Improvements and General Issues

Base PCF File Changes

BillingCenter release 3.0.6 to 3.0.7

- There are no changes to the base PCF files

Improvements to Upgrade Diff Report

With previous releases, Guidewire provided a static report in the release notes detailing certain differences between the current release and the prior minor release. This report described changes in display keys, entities, typelists, and the Gosu API. Guidewire has improved this report to dynamically conform to each customer's particular upgrade path. In addition, the report is regularly updated as the tools for generating it are enhanced to provide more information.

Because the new report is tailored to your particular upgrade requirements, it is no longer included with the general release notes. To obtain your custom Upgrade Diff Report, visit the Guidewire Resource Portal.

BillingCenter Improvements and General Issues

The following are the improvements added to and the general issues corrected in this release:

ID	Description
Commission/Commission Plan	
BC-447	Fixed a problem where overriding the commission rate for a charge with an overriding agency bill payer would fail and throw an exception, if the overriding earner and secondary earner are the same. We now check the ChargeCommission role and do not throw an exception in this case.
BC-2169	For 'On Payment Received' earning criteria, charge write-offs applied to an invoice item now earn commission for that item. Previously, that commission would be left in reserve and never become earned.
BC-2233	Fixed a problem with commission transfer (both item level, and charge level) and payment plan refactor. Previously, after payment plan refactor, the item commissions would no longer add up to the commission T-account balances. Now BillingCenter redistributes the commission across all producer codes earning on a policy period when a payment plan refactor is done.
BC-3155	The CmsnEarning batch process was updated to ensure that commission is earned only for executed payment items.
BC-3233	Fixed a problem where transferring a policy multiple times back and forth with retroactive transfer caused commission to be stuck in reserve. Commission will now correctly earn on the final producer code.
BC-4058	Fixed a problem where commission was not earned for agency bill payments after the billing method was changed to direct bill. Commission is now correctly earned.
BC-4967	Fixed a problem where overriding the commission rate for a role with no commissions was throwing a null pointer exception.
BC-8983	Fixed a problem with commission transfer and payment plan refactor intersecting. If you transferred a charge to earn on a new producer code, but then did a payment plan refactor, the item commission amounts after the payment plan refactor would not match the commission T-account balances.
BC-9042	BC-262 fixed a problem where a payment plan refactor could result in incorrect distribution of charge commission across corresponding invoice items. Now that the problem is fixed, any incorrect data can now be cleaned up as follows. Call redistributeAllCommissionForAllItems() on any charge where item commission values don't match the T-account values on their charge commissions. These charges can be identified by finding any charge with a charge commission that breaks the consistency checks in ItemCommissionDenormFieldsDBCheckBuilder: <ul style="list-style-type: none">• The ChargeCommission payable balance should equal the sum of the corresponding ItemCommission.CmsnPayableAmount values.• The ChargeCommission reserve balance should equal the sum of the corresponding ItemCommission values (CommissionAmount - CmsnPayableAmount - WrittenOffCommission - PaidCommission)• The ChargeCommission write-off balance should equal the sum of the corresponding ItemCommission.WrittenOffCommission values.• Sum of ChargeCommission expense balance and CmsnsExpenseRollup transactions should equal the sum of its ItemCommission.CommissionAmount values.

ID	Description
BC-9043	<p>Several bug fixes have been implemented to prevent creating extraneous ZeroCmsnEarnedMarkers. However, some of these could have been created in previous versions of BillingCenter. This fix provides methods to identify bad ZeroCmsnEarnedMarkers so that they can be removed.</p> <p>ZeroCmsnEarnedMarkers that should be removed are:</p> <ul style="list-style-type: none"> • ZeroCmsnEarnedMarkers on ItemCommissions that haven't had a transfer. They should only be added if commission was earned for a payment item on ONE Producer, so it shouldn't earn on other Producers. For those other producers we add them to prevent earning. If an InvoiceItem has only one ItemCommission for a role, it shouldn't have ZeroCmsnEarnedMarkers. • ZeroCmsnEarnedMarkers on ItemCommissions that aren't Pay-on-Pay. They are only relevant for that PayableCriteria. If the criteria is changed later to Pay-on-Pay, it will never earn for this payment item because of the ZeroCmsnEarnedMarker. • ZeroCmsnEarnedMarkers associated with reversed PaymentItems. Reversing payment items should have been removing ZeroCmsnEarnedMarkers. <p>Sample methods are provided to find ZeroCmsnEarnedMarkers that should be removed. Remove them by calling remove() on each suchZeroCmsnEarnedMarker. Contact support for assistance in determining the best way to do this with acceptable performance.</p> <p>See "Data fix for BC-9043: Sample code to find bad data".</p>
Database	
BC-73	<p>Fixed a problem where the database consistency check queries did not account for policy transfers from one account to another. To resolve this issue, the first check has been removed and the second check has been altered so that it does not include reversed invoice items.</p>
BC-3311	<p>Fixed a consistency check for Invoice.AmountDue.</p> <p>We now assert that Invoice.AmountDue matches the sum of unpaid balance on the items owned by the invoice, or zero if the sum is negative.</p>
BC-8748	<p>Fixed a problem where during upgrade, when consolidating PolicyCommission entities, we were not correctly updating some fields. We now are correctly updating these fields on the PolicyCommissionOfRecord.</p> <p>PrimaryPolicyPeriod is set if the resulting PolicyCommission entity is the default PolicyCommission in the primary role. The start date is set to the most recent date that we transferred to this policy commission. The end date is set to the most recent date that a policy was transferred away, or null if the PolicyCommission is the default.</p>
BC-8750	<p>In BillingCenter 3.0.5 and earlier releases, duplicate PolicyCommission entities (policy commissions with the same policy period, role and producer code) were created when transferring policy periods back to earning commission on policy commissions that they had already earned on at some point in their lifecycle.</p> <p>Beginning with BillingCenter 3.0.6 we fixed this problem and changed upgrade to consolidate these duplicates, but a problem remained with duplicate rows in the edge table bc_origchcmrn. The remaining problem was that the ChargeCommission entities being consolidated could point to two different OriginalChargeCommission entities.</p> <p>This fix solves the problem by setting the OriginalChargeCommission on the consolidated ChargeCommission entity to the most recent of the two OriginalChargeCommission entities.</p>

ID	Description
BC-8894	<p>In BillingCenter 3.0.5 and earlier releases, duplicate policy commissions (policy commissions with the same policy period, role and producer code) were created when transferring the commission earner on a policy period back to a producer code that it had previously earned on in the given role.</p> <p>Beginning with BillingCenter 3.0.6 we fixed this problem and changed upgrade to consolidate these duplicate policy commissions into a single policy commission of record. However, there were some edge case bugs with that upgrade work where certain data would cause upgrade to fail.</p> <p>One of the edge case was if there were ItemEvents where FromPolicyCommissionID or ToPolicyCommissionID pointed to a duplicate PolicyCommission that was about to be deleted. In that case we weren't correctly updating these foreign keys to the PolicyCommission of record. This left rows in the ItemEvent table that had foreign keys that pointed to entities that had been deleted from the database, causing upgrade to fail.</p> <p>We now update these foreign keys to point to the policy commission of record.</p> <p>In BillingCenter 3.0.5 and earlier releases, duplicate policy commissions (policy commissions with the same policy period, role and producer code) were created when transferring the commission earner on a policy period back to a producer code that it had previously earned on in the given role.</p> <p>Beginning with BillingCenter 3.0.6 we fixed this problem and changed upgrade to consolidate these duplicate policy commissions into a single policy commission of record. However, there were some edge case bugs with that upgrade work where certain data would cause upgrade to fail.</p> <p>One of the edge case was if there were ItemEvents where FromPolicyCommissionID or ToPolicyCommissionID pointed to a duplicate PolicyCommission that was about to be deleted. In that case we weren't correctly updating these foreign keys to the PolicyCommission of record. This left rows in the ItemEvent table that had foreign keys that pointed to entities that had been deleted from the database, causing upgrade to fail.</p> <p>We now update these foreign keys to point to the policy commission of record.</p>
Distribution	
BC-2395, BC-9030, BC-9032, BC-9034	<p>When BillingCenter makes an invoice that includes negative items billed or due, the negative items are distributed as credits as follows:</p> <ul style="list-style-type: none"> BillingCenter will use these credits to pay invoice items in order from last to first, by invoice item event date. BillingCenter will limit distribution to the amount outstanding (amount billed or due), regardless of the account's distribution limit. All other types of payment allocation and distribution (including from T-account Unapplied) remain unchanged.
Payment	
BC-9145	Fixed a problem in how BillingCenter handled reversal of written off invoice items. BillingCenter was copying the denormalized write off amount from the item being reversed, resulting in an incorrect increase in the total amount due. BillingCenter no longer copies this denormalized value onto the reversal, and the total amount due after reversing a written off item is now correct.

Data fix for BC-9043: Sample code to find bad data

See BC-9043 for more information about this fix.

The following code can be used to find bad ZeroCmsnEarnedMarker data that needs to be removed:

```

function zcemWithoutTransfers() : IQueryBeanResult<ZeroCmsnEarnedMarker> {
    var other = Query.make(ItemCommission)
    var otherPC = other.join(ItemCommission, "PolicyCommission")
    var thisZCEM = Query.make(ZeroCmsnEarnedMarker)
    var thisItemCommissions = thisZCEM.join(ZeroCmsnEarnedMarker, "ItemCommission")
    var thisPolicyCommissions = thisItemCommissions.join(ItemCommission, "PolicyCommission")

    other.compare(
        otherPC.getColumnRef("Role"),
        RelOp.Equals,
        thisPolicyCommissions.getColumnRef("Role"))
    other.compare(
        other.getColumnRef("ID"),
        RelOp.NotEquals,
        thisItemCommissions.getColumnRef("ID"))

    thisItemCommissions.subselect(

```

```

    "InvoiceItem",
    InOperation.CompareNotIn,
    other,
    "InvoiceItem")

    return thisZCEM.select()
}

function notPaymentReceivedItemLevel() : IQueryBeanResult<ZeroCmsnEarnedMarker> {
    return Query.make(ZeroCmsnEarnedMarker)
        .join(ZeroCmsnEarnedMarker, "ItemCommission")
        .compare("PayableCriteria", Relop.NotEquals, PayableCriteria.TC_PAYMENTRECEIVED)
        .compare("PayableCriteria", Relop.NotEquals, null)
        .select()
}

function notPaymentReceivedChargeLevel() : IQueryBeanResult<ZeroCmsnEarnedMarker> {
    var zcem = Query.make(ZeroCmsnEarnedMarker)
    var ic = zcem.join(ZeroCmsnEarnedMarker, "ItemCommission")
    ic.compare("PayableCriteria", Relop.Equals, null)
    var ii = ic.join(ItemCommission, "InvoiceItem")
    var c = ii.join(InvoiceItem, "Charge")
    var cc = c.join(ChargeCommission, "Charge")
    zcem.compare(
        cc.getColumnRef("PolicyCommission"),
        Relop.Equals,
        ic.getColumnRef("PolicyCommission"))
    cc.compare(
        "PayableCriteria",
        Relop.NotEquals,
        PayableCriteria.TC_PAYMENTRECEIVED)
    cc.compare("PayableCriteria", Relop.NotEquals, null)
    return zcem.select()
}

function notPaymentReceivedPolicyLevel() : IQueryBeanResult<ZeroCmsnEarnedMarker> {
    var zcem = Query.make(ZeroCmsnEarnedMarker)
    var ic = zcem.join(ZeroCmsnEarnedMarker, "ItemCommission")
    ic.compare("PayableCriteria", Relop.Equals, null)
    var ii = ic.join(ItemCommission, "InvoiceItem")
    var c = ii.join(InvoiceItem, "Charge")
    var cc = c.join(ChargeCommission, "Charge")
    cc.compare("PayableCriteria", Relop.Equals, null)
    var pc = cc.join(ChargeCommission, "PolicyCommission")
    var subplan = pc.join(PolicyCommission, "CommissionSubPlan")
    zcem.compare(
        ic.getColumnRef("PolicyCommission"),
        Relop.Equals,
        pc.getColumnRef("ID"))
    zcem.compare(
        cc.getColumnRef("PolicyCommission"),
        Relop.Equals,
        pc.getColumnRef("ID"))
    subplan.compare("PayableCriteria", Relop.NotEquals, PayableCriteria.TC_PAYMENTRECEIVED)
    subplan.compare("PayableCriteria", Relop.NotEquals, null)

    return zcem.select()
}

private function zcemOnReversedPaymentItems() : IQueryBeanResult<ZeroCmsnEarnedMarker> {
    return Query.make(ZeroCmsnEarnedMarker)
        .join(ZeroCmsnEarnedMarker, "BasePaymentItem")
        .compare("ReversedDate", Relop.NotEquals, null)
        .select()
}

```

Platform Improvements and General Issues

ID	Description
Batch Processes	

PL-14589	Guidewire has implemented a new batch process, <code>ProcessCompletionMonitor</code> . The process wakes up at user-configurable intervals and examines the Process History and Work Queue for all batch processes. After the batch process completes, it invokes plugin <code>IBatchCompletedNotification</code> to permit user code to react to the completion of the batch process. This process also marks <code>ProcessHistory.NOTIFICATIONSENT = true</code> for that batch process to ensure that the plugin is invoked only once. Note that you must provide an implementation of plugin <code>IBatchCompleteNotification</code> .
Code Utilities	
PL-17761	Corrected the numerical count of active sessions for active users. Previously, active sessions were being timed out for the count, even though users were still active.
Configuration Environment	
PL-13071	Added the ability to verify PCF files outside of Guidewire Studio through the use of the <code>gwbc verify-types</code> command line utility.
Consistency Checker	
PL-16620	<p>Database parameter <code>checker.threads</code> (in <code>config.xml</code>) controls the number of threads used to execute database consistency checks (DBCC). Previously, you could only control the number of threads using the synchronous methods of running DBCC, from the <code>system_tools -checkdbconsistency</code> command line and at application start-up. With this change, you can override the number of threads if you are running consistency checks as a batch job.</p> <ul style="list-style-type: none"> If you set a value for <code>checker.threads</code> in <code>config.xml</code>, then BillingCenter displays the configured number of threads on the (System Tools) Info Page → Consistency Checks page. You can override the configured value with a positive integer and submit new consistency checks with this new value on that page. If you do not set a value for <code>checker.threads</code> in <code>config.xml</code>, then BillingCenter uses 1 as the default value and displays that value on the Consistency Checks page. You can still override this value on that page.
Database	
PL-16762	Oracle. Guidewire has added a new <code><database></code> parameter in <code>config.xml</code> that you use—in Oracle-based implementations—to disable rewrite queries using materialized views at a session level. The name of the parameter is <code>queryRewriteEnabled</code> . Set its value to <code>false</code> to disable rewrite queries.
PL-17133	Fixed an issue in which BillingCenter would open a database connection and never close it if accessing an external data source using JNDI. This prevented database fail-over. This fix changes this behavior so BillingCenter closes all connections if they are no longer in use.
Queries	
PL-12485	Guidewire has deprecated the <code>RawQuery</code> property. Instead, create a <code>gw.api.filters.StandardQueryFilter</code> and use its <code>filterQuery</code> method or the <code>IQueryBeanResult.addFilter</code> method to apply the filter to a query.
Upgrade	
PL-18021	Fixed an issue in which the Upgrade Tool did not handle rules upgrade properly
Web	
PL-16213	Modified the way that the application renders HTML to include a <code>LABEL</code> element for each of the different types of inputs in a form in order to support assistive technologies such as a screen reader.
PL-16711	Improved performance by modifying the UI-rendering JavaScript so that it now only resizes visible widgets.
XML	
PL-16706	Update to allow multi-threading of date formatting for File Based Archiving to improve performance.

Known Issues and Limitations

This section describes known issues with this release of Guidewire BillingCenter.

- BillingCenter Known Issues
- Platform Known Issues

BillingCenter Known Issues

[Upgrade Guide: Typo on the Supported Starting Version page \(BC-9137\)](#)

Issue: The BillingCenter Upgrade Guide states that you can upgrade from any 2.1.x version.

However, BillingCenter 2.1.7 is the earliest version you can upgrade from. If you are upgrading from a 2.1.x version earlier than 2.1.7, Guidewire recommends that you first upgrade to the latest 2.1.x version and then upgrade to the latest 3.0.x version of BillingCenter.

Workaround: If you need to upgrade from a prior version, consult Guidewire Services for assistance.

[Commissions are calculated for non-commissionable items when overriding the commission rate \(BC-9083\)](#)

Issue: When an agency bill policy has a commission override, the commission amount incorrectly includes non-commissionable items.

Workaround: Override the commission rate at charge level, instead of for the policy. To override commission at the charge level programmatically:

1. Call Charge.overrideCommissionRate(PolicyRole policyRole, BigDecimal newCommissionRate) on each commissionable charge.
2. Commit the bundle.

[Payments against Pay-on-Pay Policies with 0% rate can never be re-earned \(BC-1270\)](#)

Issue: Payments on policies with commission rate of 0% and commission earning criteria of pay on pay can not be re-earned if the commission rate is later changed to a non zero rate. For example:

Create an agency bill policy, with commission set to pay on pay, 0% commission rate, \$1000 premium

1. Fully pay policy using direct bill payment.
2. Override Commission Rate on Charge to 10%.
3. Change the policy's billing method to direct bill.
4. Run the CmsnPayable batch process.

Expected: \$100 commission is payable to the producer.

Actual: The commission remains in reserve and does not become payable.

Workaround: Commission for the producer on the primary role can be earned by setting the payable criteria on the invoice commission item to earn on bind. Payable criteria can be set to earn on bind by using the following gosu code:

```
var policyPeriod = find (pp in PolicyPeriod where pp.PolicyNumber == "test2").AtMostOneRow
print(policyPeriod)

var adjustmentItem = find (ii in InvoiceItem where ii.PolicyPeriod == policyPeriod AND ii.Type
== InvoiceItemType.TC_COMMISSIONADJUSTMENT).AtMostOneRow
print (adjustmentItem)

gw.transaction.Transaction.runWithNewBundle(\ bundle -> {
    adjustmentItem = bundle.add(adjustmentItem)
    adjustmentItem.ActivePrimaryItemCommission.PayableCriteria = PayableCriteria.TC_BINDING
    bundle.commit()
})
```

On AIX without an X11 server, BillingCenter may report that an X11 resource is missing (CC-5889)

Issue: BillingCenter does not require an X11 server; however other components of the BillingCenter environment (such as Java) may inadvertently have this requirement. If so, BillingCenter may report that an X11 resource is missing when it is starting up.

Workaround: Install an X11 server, or contact Guidewire support to determine if the dependency can be removed.

Incorrect additivity statements in the default *logging.properties* file (CC-50615)

Issue: Some additivity statements in the `logging.properties` file are incorrect and therefore, ineffective. An additivity statement attempts to keep the category from sending its log messages up to its parent, preventing duplicate log messages across files.

Workaround: In the `logging.properties` file (located at `modules/cc/config/logging`), change:

```
log4j.additivity.PluginsLog=false  
to  
log4j.additivity.Plugin=false
```

Problems occurred with the use of negative collateral charges (CC-51311)

Issue: When issuing a negative collateral charge, the amount gets subtracted from a Charge Due T-account and a payer's Unapplied T-account. This causes an imbalance between `collateral_reserve` and `collateral_held` when issuing a negative charge.

Workaround: Guidewire is aware of this issue.

Automatic distribution to policies not working correctly (BC-8633, was CC-51511)

Issue: Automatic distribution to policies on **Account → Disbursement** screen does not distribute the payment correctly.

Workaround: Guidewire is aware of this issue.

Billing instructions for Issuance and Policy Change must work with premium and other charges (Premium Reporting Only) (CC-52032)

Issue: This issue is for Premium Reporting policy periods only. Users are not allowed to enter premium charges as part of an issuance or policy change and process them as any other policy period. This issue applies to all billing instructions types.

Workaround: Guidewire is aware of this issue.

No commission reserve and onset transfer transaction was created when a direct bill item was reassigned to an agency bill producer (CC-54923)

Issue: When a direct bill policy has no specified primary producer and one of the policy's item is reassigned to an agency bill producer, no commission reserve and onset transfer transaction created. This was because there is no initial reserve.

Workaround: Guidewire is aware of this issue.

New collateral charges and invoice amounts are being generated based on previously paid collateral amounts, not the previously billed or planned amounts (CC-55209)

Issue: New collateral charges and invoice amounts are being generated based on the amount of collateral previously paid and do not account for amounts of collateral previously Billed or Planned (not yet billed). This results in an over billing of the collateral amount on the invoices.

Workaround: Guidewire is aware of this issue.

[Errors occur in the totals for the producer's rate on the Transactions screen \(CC-55835\)](#)

Issue: A difference occurs in the totals shown on the **Transactions** screen for a producer when selecting **Reserves** and then choosing either the **Last 30 Days** or **Show All** filter.

Workaround: Guidewire is aware of this issue.

[Account Evaluation is not counting account level delinquencies \(CC-55938\)](#)

Issue: The count for number of delinquencies on **Account Evaluation** screen does not factor account level delinquencies.

Workaround: Use the Gosu finder to identify the account level delinquency processes and add it to the BillingCenter value.

[After a reversal, the distributed payment is still being reserved for the premium payment \(CC-56538\)](#)

Issue: The disbursement amount is not fully moved to disbursement and does not show up on the **Disbursement** page although an entire payment has been reversed and the associated charge had been assigned to another account.

Workaround: This is a new feature that is not fully implemented. Guidewire is aware of this issue.

[Moving a Billed invoice item should negate the direct bill commission paid, not the agency bill commission paid \(CC-56656\)](#)

Issue: If an item is moved from a **Billed** invoice to a **Planned** invoice, an earnings transaction is created for the Payable amount on the commission. However, if the commission was earned and paid, it is not being negated.

Workaround: Guidewire is aware of this issue.

[Incorrect totals in the policy payment schedule occurred on agency bill statements after a change was made to one of two producers on a policy \(CC-56925\)](#)

Issue: The producer statement does not show the correct dollar amount after a change is made to one of the policy's payment plans on a policy with two producers after the statement was billed and partially paid.

Workaround: Guidewire is aware of this issue.

[Should not be able to edit Payment Plan after it has been used by a policy \(CC-57758\)](#)

Issue: In the base configuration, it is possible to edit a Payment Plan even after it has been used by a policy. This is incorrect. Only the expiration date should be updateable.

Workaround: Configure the payment plan screen to only allow the expiration date to be updated.

[Invoice PreUpdate rule is bypassed when a new invoice created for Collateral charge \(CC-58254\)](#)

Issue: Automatic distribution to policies on **Account → Disbursement** screen does not distribute the payment correctly.

Workaround: Guidewire is aware of this issue.

[No API to remove invoice items from DirectBillPaymentView \(CC-58397\)](#)

Issue: There is no way to programmatically remove an Invoice item (or invoice item array) from a distribution.

Workaround: Guidewire is aware of this issue.

[Use of StringBuilder Class prevents debugging of web service \(CC-58533\)](#)

Issue: In a web service, if you have a method with a parameter of type X and inside the web service method you reference a Gosu class of type Y. And, Gosu class Y has a `toString()` method (overriding the default) that uses a `java.lang.StringBuilder` class. If you set a breakpoint in the web service method and try to step through the method, then you will get errors.

Workaround: Guidewire is aware of this issue.

[Reversing Write-Off Commissions and Write-Off Commission Recovery in the UI \(CC-58655\)](#)

Issue: The BillingCenter user interface does not provide a way reverse Write-Off Commissions and Write-Off Commission Recovery.

Workaround: Clone the `Reverse` button under the `Payments` tab and reverse `CmsnPayableReduction` and other transactions using the following action: `CurrentLocation.startEditing(); producerTransaction.reverse();` and `CurrentLocation.commit();`.

[Cannot un-distribute direct bill payment \(CC-58764, CC-58764\)](#)

Issue: The base configuration does not allow a direct bill payment to be undistributed.

Workaround: Use the configuration to create an `Unapply` button on the `Payment List View`. This button should remove the original `DirectBillMoneyRcvd`, reverse all associated transactions, and then create a brand new `DBMoneyRcvd` with the same attributes as the original.

[List view on New Account and New Producer wizard does not display properly \(CC-59574\)](#)

Issue: Attempting to add an existing contact following a new contact causes the added contact to not display properly in the list view on the `Add Contact` wizard on the `New Account` and `New Producer` screens.

Workaround: Click on a column heading and the sort cause the screen to refresh and display the contacts correctly.

[Rewrite BI with the same policy number as the original policy is not incrementing the TermNumber and throws DBDuplicateKeyException \(CC-59691\)](#)

Issue: Sending a rewrite billing instruction without specifying the term number will throw a `DBDuplicateKeyException` if term number “1” already exists for the policy. If you omit the term number, BillingCenter uses the number “1” for term number. If a policy period with the term number 1 already exists then BillingCenter will throw an exception.

Workaround: Include a term number that is unique for the given policy number when sending a rewrite billing instruction.

[Agency Bill Payment wizard generates `EvaluationException` \(CC-59721\)](#)

Issue: Currently, if you click `Execute` on the `Execute Agency Promise As Payment` screen to apply a payment to a promise that has been pre-filled with zeroes, the following exception occurs:

`EvaluationException: Cannot re-add AgencyCyclePayment since it has already been removed.`

Workaround: Use the Payment Wizard to apply the payment instead.

Platform Known Issues

Issues with Internet Explorer 9

Issue: If you are using the Internet Explorer 9 browser, it is possible to see issues such as screen flickering or an incorrect tab order for fields. According to public reports, IE 9 exhibits these and other issues with a variety of web sites and web applications.

Workaround: Because this is the behavior of the Internet Explorer 9 rendering engine, Guidewire cannot address these issues. However, there are reports of IE 9 users being able to reduce these issues by changing the new IE 9 Accelerated Graphics settings on the Advanced tab of the [Internet Options](#) dialog.

First time you click on the arrow of the typekey input, the drop-down menu will not open (PL-10134)

Issue: The drop-down menu does not open on the first click of the arrow on a typekey input. Instead, the help text opens.

Workaround: Turn off help text on focus by setting `InputHelpTextOnFocus` to `false` in the `config.xml` file. By doing that, the help text shows only if you mouse over the input and will not interfere with opening a drop down menu.

Studio Rules do not use correct capitalization for root object's name (PL-10740)

Issue: Rule set root objects are not named with first letter lower-cased.

Workaround: Rules engine will issue warnings when correct case for objects is not being used.

Geocode plugin calls 'isSufficientlyCompleteToGeocode' method twice for each address (PL-11578)

Issue: The Geocode plugin calls the `isSufficientlyCompleteToGeocode` method twice for each address.

Workaround: Guidewire is aware of this issue.

Countries configured in 'zone-config.xml' still generate a warning during regen-dictionary even when zone data is loaded for all of these countries (PL-11947)

Issue: Countries configured in `zone-config.xml` still generate a warning during regen-dictionary even when zone data is loaded for all of these countries.

Workaround: Warning message is created in error and can safely be ignored.

Type system refresh after PCF page title change does not update corresponding menu label (PL-13057)

Issue: The type system refresh after the PCF page title change does not update corresponding menu label.

Workaround: After updating a page title, the server restart must be done to refresh menu labels and avoid null pointer exceptions due to stale reference.

Length limitation on entity localization table names (PL-13360)

Issue: There is a length limitation on entity localization table names.

Workaround: Ensure that localization `tableName` property specified in the entity extension file is less than 16 characters. The error message generated if the localization table name exceeds the maximum length indicates that 18 characters are allowed, but that does not account for two additional characters added by the application.

'US-Locations.txt' file with the U.S. geodata from GreatData has special characters that cause validation problems with United States Postal Service (USPS) data (PL-13384)

Issue: The `US-Locations.txt` file contains information that does not conform to United States Postal Service (USPS) standards for bulk mailings.

Workaround: The provided `US-Locations.txt` file is intended only for use in geocoding to identify addresses for a location. You can process the `US-Locations.txt` file to conform to your particular address standards, and then import that version of the file instead.

GX models that reference virtual fields and enhancements throw null pointers if null (PL-13560)

Issue: The GX models that reference virtual fields and enhancements throw null pointers when they are null.

Workaround: Ensure that null checks and error handling is included so that if referenced virtual fields or enhancements are null, then there will not be a null pointer exception.

Email with file attachment with unicode filename not correctly handed over to the mail server (PL-13582)

Issue: Sending email with file attachment with unicode filename is not sent to the mail server correctly.

Workaround: Use Latin characters for file names on attached files.

The GX model generated XSD cannot be parsed by JAXB (PL-13598)

Issue: The GX model generated XSD cannot be parsed by JAXB.

Workaround: Add JAXB annotation elements to the XSD to specify the necessary metadata (such as class names) for JAXB to generate the java class files. Contact Guidewire Support for an example XSD annotated in this way.

'JavaToolkit.gs' has incorrectly hard coded memory which results in failed 'regen-java-api' Ant task (PL-13663)

Issue: The `JavaToolkit.gs` has hard coded memory which can result in failed `regen-java-api` Ant tasks.

Workaround: Increase the size of the maximum heap setting on line 161 of `JavaToolkit.gs` in the Ant module. The default value is 512.

Cannot make a delegate field a localized column (PL-13761)

Issue: You cannot make a delegate field a localized column.

Workaround: Move the column that you want to localize off the delegate and onto each of the implementing entities. Then, in order to make the column appear as though it exists on the delegate, define an enhancement property on the delegate, that *delegates* to the appropriate column, depending on the implementing entity.

Cannot print second-level ListView (PL-14640)

Issue: Printing a second-level list view does not work correctly. The print job contains duplicates and does not include all of the items.

Workaround: Remove the lower level list view out of the hierarchical structure and include it at the top level.

RowSet and RowIterator configured in the same ListView do not work correctly. (PL-14946)

Issue: Configuring a RowSet and RowIterator in the same ListView does not provide the right behavior for editability.

Workaround: Separate the RowSet and RowIterator that exist in the same ListView into separate components or into multiple ListView widgets.

[Masked input fields do not correctly handle ENTER keypress \(PL-14955\)](#)

Issue: Input fields that have a field validator mask do not correctly handle the ENTER keypress. After you press ENTER, the field no longer displays the input mask and it is not possible to enter any further input. This is an issue with Microsoft Internet Explorer.

Workaround: Cancel the page and re-enter the information.

[BillingCenter does not recognize modified 'Calendar.js' file \(PL-14997\)](#)

Issue: BillingCenter does not correctly recognize changes made to `Calendar.js` through Studio. You access this file in the following Studio location:

`Resources → Web Resources → resources → javascript → global`

Workaround: Instead, modify `global.js` directly. You access this file in the following installation file location:

`BillingCenter/webapps/bc/resources/javascript`

WARNING This is one of the very rare occasions in which you modify an application file outside of the application configuration folder. Ensure that you modify `global.js` only. Otherwise, you can invalidate your installation.



chapter 30

Guidewire BillingCenter 3.0.6 Release Notes

Release 3.0.6.3

Document Revision: 02-July-2015

Copyright © 2001-2015 Guidewire Software, Inc. All rights reserved.

Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, Guidewire BillingCenter, Guidewire Reinsurance Management, Guidewire ContactManager, Guidewire Vendor Data Management, Guidewire Client Data Management, Guidewire Rating Management, Guidewire InsuranceSuite, Guidewire ContactCenter, Guidewire Studio, Guidewire Product Designer, Guidewire Live, Guidewire DataHub, Guidewire InfoCenter, Guidewire Standard Reporting, Guidewire ExampleCenter, Guidewire Account Manager Portal, Guidewire Claim Portal, Guidewire Policyholder Portal, Gosu, Deliver Insurance Your Way, and the Guidewire logo are trademarks, service marks, or registered trademarks of Guidewire Software, Inc. in the United States and/or other countries.

All other trademarks are the property of their respective owners.

This material is confidential and proprietary to Guidewire and subject to the confidentiality terms in the applicable license agreement and/or separate nondisclosure agreement.

Guidewire products are protected by one or more United States patents.

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire BillingCenter.

Version Number

This release of Guidewire BillingCenter is 3.0.6.3.

For Standard Reporting, this release supports InetSoft StyleReport Enterprise Edition 10.1, 20100113.

Support

For assistance, visit the Guidewire Resource Portal – <http://guidewire.custhelp.com>

Installing This Release

Refer to the following:

- The *BillingCenter Installation Guide* for general installation information
- The *BillingCenter Reporting Guide* if you plan to use the integrated reporting feature
- BillingCenter release notes previous releases for any versions that have been skipped

InetSoft Evaluation

You can evaluate InetSoft Style Report Enterprise Edition Version 10.1 using the following license key:

L000-72C-ERX-0000BF00100001F-F2B025280327

If you want to use InetSoft and the BillingCenter reporting module in a production environment, contact Guidewire Customer Support to request a full license key. For more information, refer to the *BillingCenter Reporting Guide* documentation.

WARNING The build number of the reporting install files must match the build number of the Guidewire BillingCenter installation. If the build numbers do not match, you will have problems with your reporting installation.

Changes in this Release

This section describes the product changes in this release.

- Base PCF File Changes
- BillingCenter Improvements and General Issues

Base PCF File Changes

All links below require the `ReleaseNotes_files` directory on your local disk.

BillingCenter release 3.0.5 to 3.0.6

- There were no changes to the base PCF files in the `modules/bc` directory.
- To view a report of the changes in the base PCF files in the `modules/p1` directory, [click here](#).

BillingCenter Improvements and General Issues

The following are the improvements added to and the general issues corrected in this release:

ID	Description
Commission/Commission Plan	
CC-59692	<p>Before this fix, when running multiple workers for commission payable batch process, not all the commission items got earned. This was because the CustomPayableCalculator contains its own state, which needs to be set per worker.</p> <p>This fix creates a new CustomPayableCalculator instance for each worker to avoid state corruption.</p>
CC-59772	Fixed a problem where the commission payable batch process did not make all commission become earned after multiple retroactive policy transfers.
Database	
CC-59374	<p>Before this fix we based producer payments on the transactions for an item instead of the CmsnPayableAmount field. The UpgradeCommissions batch process was setting this value without creating any transactions on that item, which caused inconsistency. The ProducerPayment batch now needs the item-level transactions to be there, so we have to create them.</p> <p>Now the only earning and payable transactions that should exist in our system without an InvoiceItemID are reversed and reversal.</p>
CC-60053	Upgrade: Fixed upgrade of NegativeChargeBilled transactions so that all the line items are moved over correctly.
Disbursement	
CC-59732	<p>Before this fix the AutomaticDisbursement batch process would include commission in the amount to disburse.</p> <p>Now the AutomaticDisbursement batch process correctly computes the amount of the disbursement, as follows: Account Unapplied minus total gross unpaid amount of invoice items on the account.</p>
Gosu	
CC-60102	Previously, the GeneralUtil.gs class was coupled to the BaseCommand.gs class, meaning that it could not be used in GUnit code. This has been fixed and GeneralUtil.gs can now be used in the GUnit test environment.
Integration	
CC-59838	This fix addresses a problem that occurred when BillingCenter executed a renewal billing instruction. If the offer number was set, BillingCenter would try to reverse all suspense payments with the same offer number, even if they were already reversed.
Miscellaneous	
CC-60062	Added a consistency check to ensure each transaction is balanced, meaning that the total credit line items is equal the total debit line items.
Performance	
CC-59973	This fix improves the performance and scalability of the ProducerPayment batch process. BillingCenter no longer creates zero dollar commission reserve transactions for non-commissionable charges.
CC-59535	This fix improves conversion performance by doing row numbering only once, at the end of the process.
CC-59684	This fix improves conversion performance by removing unnecessary SQL statement joins.
CC-60009	<p>This fix improves ProducerPayment performance by only committing InvoiceItems that have been modified.</p> <p>Note: No changes are necessary to maintain current performance. To take full advantage of this fix and achieve optimal performance, the number of configured worker instances or batchsize may need to be adjusted.</p>
Payment	
CC-59776	<p>Before this fix, modifying an item other than the suspense items on a payment with suspense items would result in the suspense items not being included in the resulting net amount distributed.</p> <p>Now the net amount distributed is calculated correctly by including the net amount of the suspense items.</p>

ID	Description
Producer	
CC-59016	<p>The ProducerPayment batch process now acts like it was run on the date it was scheduled to (<code>ProducerRecurable.NextDate</code>), even if it runs late.</p> <p>This means:</p> <ul style="list-style-type: none"> The statement will be stamped (<code>StatementDate</code> field) with the scheduled date, and <i>not</i> the actual date the batch was run. If multiple scheduled dates pass before the batch process is run, the batch process will create a statement for each scheduled date, and place transactions executed for each statement period on the corresponding statement. Associated outgoing payments will have the <code>Amount</code> field initialized to the amount that will be paid on the <code>ProducerPaymentSent</code> transaction. This change will not affect existing configurations. Charge-level commission will roll up to the producer only if it was earned on or before the scheduled date. It will consequently not contribute to the auto-payment if it was not rolled up. Policy Incentive-based bonuses will roll up to the producer level only if they were earned on or before the scheduled date. Producer Code bonuses scheduled for Next Payment will be executed only if they entered the system on or before the scheduled date. (Note that bonuses executed Immediately are earned, rolled up, and paid immediately, so nothing awaits the batch process in this case.) Any Producer-level advances scheduled for Next Payment will be executed only if they entered the system on or before the scheduled date. Any Producer-level advances that were executed before the batch runs will be repaid only if they were executed before the scheduled date (and all other normal conditions are met).
Transaction	
CC-59991	This fix updates the ProducerPayment batch process to not repay advance if it would bring payable to negative. The fix also prevents ProducerPayment from issuing payments that would make payable become negative.

Platform Improvements and General Issues

Core	
PL-12485	Guidewire no longer exposes internal implementation details of a query through the undocumented <code>RawQuery</code> property of the query builder APIs. Guidewire has removed this property.
PL-13071	Added the ability to verify PCF files outside of Guidewire Studio through the use of <code>gwbc verify-types</code> command line utility.
PL-14589	Guidewire has implemented a new batch process, <code>ProcessCompletionMonitor</code> . The process wakes up at user-configurable intervals and examines the Process History and Work Queue for all batch processes. After the batch process completes, it invokes user plugin <code>IBatchCompletedNotification</code> to permit user code to react to the completion of the batch process. This process also marks <code>ProcessHistory.NOTIFICATIONSENT = true</code> for that batch process to ensure that the plugin is invoked only once.
PL-16856	Created a database consistency check to detect certain problems with <code>EffDated</code> localization entities: <ul style="list-style-type: none"> Out-of-sequence changes that result in multiple entries for the same language. Gaps in the revision history for localization entities. Setting the value to <code>null</code> causes gaps in the revision history. In general, gaps in the revision history are acceptable for <code>EffDated</code> entities, but not they are not acceptable for localization entities.
PL-16857	Fixed a problem in which out-of-sequence changes to localized columns caused data inconsistencies for non-default locales. This fix changes how BillingCenter handles localized columns in diff reports. Diffs now report changes in non-default locales as normal property changes.

Database	
PL-16620	<p>Database parameter checker.threads (in config.xml) controls the number of threads used to execute database consistency checks (DBCC). Previously, you could only control the number of threads using the synchronous methods of running DBCC, from the system_tools -checkdbconsistency command line and at application start-up. With this change, you can override the number of threads if you are running consistency checks as a batch job.</p> <p>If you set a value for checker.threads in config.xml, then BillingCenter displays the configured number of threads on the (System Tools) Info Page → Consistency Checks page. You can override the configured value with a positive integer and submit new consistency checks with this new value on that page.</p> <p>If you do not set a value for checker.threads in config.xml, then BillingCenter uses 1 as the default value and displays that value on the Consistency Checks page. You can still override this value on that page.</p>
PL-16762	Guidewire has added a new <database> parameter in config.xml that you use—in Oracle-based implementations—to disable rewrite queries using materialized views at a session level. The name of the parameter is queryRewriteEnabled. Set its value to false to disable rewrite queries.
PL-16763	<p>Added a new configuration parameter—QueryRewriteForClaimSearch—that enables options for query rewrite on Oracle. There are options to rewrite queries using materialized view based on cost or to force the rewrite. The four possible values are as follows:</p> <ul style="list-style-type: none"> • <param name="QueryRewriteForClaimSearch" value="COST/NOSTALE"/> • <param name="QueryRewriteForClaimSearch" value="COST/STALE"/> • <param name="QueryRewriteForClaimSearch" value="FORCE/NOSTALE"/> • <param name="QueryRewriteForClaimSearch" value="FORCE/STALE"/> <p>If you provide an invalid value, the server ignores it.</p>
PL-17133	Fixed an issue in which BillingCenter would open a database connection and never close it if accessing an external data source using JNDI. This prevented database fail-over. This fix changes this behavior so BillingCenter closes all connections if they are no longer in use.
Integration	
PL-13499	Implemented a BingMaps plugin as a Geocode plugin to replace Mappoint, as Microsoft intends to discontinue that service in November of 2011.
PL-16706	Update to allow multi-threading of date formatting for File Based Archiving to improve performance.
Web	
PL-16711	Improved performance by modifying the UI-rendering JavaScript so that it now resizes only visible widgets.

Known Issues and Limitations

Note: For maintenance releases, it is Guidewire policy to defer fixing configuration issues if they require merging files during the upgrade. Workarounds to many of these issues are listed in the following sections. The goal of this policy is to make upgrades as straightforward as possible.

This section describes known issues with this release of Guidewire BillingCenter:

- BillingCenter Issues
- Platform Issues

BillingCenter Issues

On AIX without an X11 server, BillingCenter may report that an X11 resource is missing (CC-5889)

Issue: BillingCenter does not require an X11 server; however other components of the BillingCenter environment (such as Java) may inadvertently have this requirement. If so, BillingCenter may report that an X11 resource is missing when it is starting up.

Workaround: Install an X11 server, or contact Guidewire support to determine if the dependency can be removed.

Incorrect additivity statements in the default *logging.properties* file (CC-50615)

Issue: Some additivity statements in the `logging.properties` file are incorrect and therefore, ineffective. An additivity statement attempts to keep the category from sending its log messages up to its parent, preventing duplicate log messages across files.

Workaround: In the `logging.properties` file (located at `modules/cc/config/logging`), change:

```
log4j.additivity.PluginsLog=false  
to  
log4j.additivity.Plugin=false
```

Problems occurred with the use of negative collateral charges (CC-51311)

Issue: When issuing a negative collateral charge, the amount gets subtracted from a Charge Due T-account and a payer's Unapplied T-account. This causes an imbalance between `collateral_reserve` and `collateral_held` when issuing a negative charge.

Workaround: Guidewire is aware of this issue.

Automatic distribution to policies not working correctly (CC-51511)

Issue: Automatic distribution to policies on `Account → Disbursement` screen does not distribute the payment correctly.

Workaround: Guidewire is aware of this issue.

Billing instructions for Issuance and Policy Change must work with premium and other charges (Premium Reporting Only) (CC-52032)

Issue: This issue is for Premium Reporting policy periods only. Users are not allowed to enter premium charges as part of an issuance or policy change and process them as any other policy period. This issue applies to all billing instructions types.

Workaround: Guidewire is aware of this issue.

No commission reserve and onset transfer transaction was created when a direct bill item was reassigned to an agency bill producer (CC-54923)

Issue: When a direct bill policy has no specified primary producer and one of the policy's item is reassigned to an agency bill producer, no commission reserve and onset transfer transaction created. This was because there is no initial reserve.

Workaround: Guidewire is aware of this issue.

New collateral charges and invoice amounts are being generated based on previously paid collateral amounts, not the previously billed or planned amounts (CC-55209)

Issue: New collateral charges and invoice amounts are being generated based on the amount of collateral previously paid and do not account for amounts of collateral previously `Billed` or `Planned` (not yet billed). This results in an over billing of the collateral amount on the invoices.

Workaround: Guidewire is aware of this issue.

Errors occur in the totals for the producer's rate on the Transactions screen (CC-55835)

Issue: A difference occurs in the totals shown on the `Transactions` screen for a producer when selecting `Reserves` and then choosing either the `Last 30 Days` or `Show All` filter.

Workaround: Guidewire is aware of this issue.

[Account Evaluation is not counting account level delinquencies \(CC-55938\)](#)

Issue: The count for number of delinquencies on **AccountEvaluation** screen does not factor account level delinquencies.

Workaround: Use the Gosu finder to identify the account level delinquency processes and add it to the BillingCenter value.

[After a reversal, the distributed payment is still being reserved for the premium payment \(CC-56538\)](#)

Issue: The disbursement amount is not fully moved to disbursement and does not show up on the **Disbursement** page although an entire payment has been reversed and the associated charge had been assigned to another account.

Workaround: This is a new feature that is not fully implemented. Guidewire is aware of this issue.

[Moving a Billed invoice item should negate the direct bill commission paid, not the agency bill commission paid \(CC-56656\)](#)

Issue: If an item is moved from a **Billed** invoice to a **Planned** invoice, an earnings transaction is created for the Payable amount on the commission. However, if the commission was earned and paid, it is not being negated.

Workaround: Guidewire is aware of this issue.

[Incorrect totals in the policy payment schedule occurred on agency bill statements after a change was made to one of two producers on a policy \(CC-56925\)](#)

Issue: The producer statement does not show the correct dollar amount after a change is made to one of the policy's payment plans on a policy with two producers after the statement was billed and partially paid.

Workaround: Guidewire is aware of this issue.

[Should not be able to edit Payment Plan after it has been used by a policy \(CC-57758\)](#)

Issue: In the base configuration, it is possible to edit a Payment Plan even after it has been used by a policy. This is incorrect. Only the expiration date should be updateable.

Workaround: Configure the payment plan screen to only allow the expiration date to be updated.

[Invoice PreUpdate rule is bypassed when a new invoice created for Collateral charge \(CC-58254\)](#)

Issue: Automatic distribution to policies on **Account → Disbursement** screen does not distribute the payment correctly.

Workaround: Guidewire is aware of this issue.

[No API to remove invoice items from DirectBillPaymentView \(CC-58397\)](#)

Issue: There is no way to programmatically remove an Invoice item (or invoice item array) from a distribution.

Workaround: Guidewire is aware of this issue.

Use of StringBuilder Class prevents debugging of web service (CC-58533)

Issue: In a web service, if you have a method with a parameter of type X and inside the web service method you reference a Gosu class of type Y. And, Gosu class Y has a `toString()` method (overriding the default) that uses a `java.lang.StringBuilder` class. If you set a breakpoint in the web service method and try to step through the method, then you will get errors.

Workaround: Guidewire is aware of this issue.

Reversing Write-Off Commissions and Write-Off Commission Recovery in the UI (CC-58655)

Issue: The BillingCenter user interface does not provide a way reverse Write-Off Commissions and Write-Off Commission Recovery.

Workaround: Clone the Reverse button under the Payments tab and reverse `CmsnPayableReduction` and other transactions using the following action: `CurrentLocation.startEditing(); producerTransaction.reverse();` and `CurrentLocation.commit();`.

Cannot un-distribute direct bill payment (CC-58764, CC-58764)

Issue: The base configuration does not allow a direct bill payment to be undistributed.

Workaround: Use the configuration to create an `Unapply` button on the Payment List View. This button should remove the original `DirectBillMoneyRcvd`, reverse all associated transactions, and then create a brand new `DBMoneyRcvd` with the same attributes as the original.

List view on New Account and New Producer wizard does not display properly (CC-59574)

Issue: Attempting to add an existing contact following a new contact causes the added contact to not display properly in the list view on the `Add Contact` wizard on the `New Account` and `New Producer` screens.

Workaround: Click on a column heading and the sort cause the screen to refresh and display the contacts correctly.

Rewrite BI with the same policynumber as the original policy is not incrementing the TermNumber and throws DBDuplicateKeyException (CC-59691)

Issue: Sending a rewrite billing instruction without specifying the term number will throw a `DBDuplicateKeyException` if term number “1” already exists for the policy. If you omit the term number, BillingCenter uses the number “1” for term number. If a policy period with the term number 1 already exists then BillingCenter will throw an exception.

Workaround: Include a term number that is unique for the given policy number when sending a rewrite billing instruction.

Agency Bill Payment wizard generates `EvaluationException` (CC-59721)

Issue: Currently, if you click `Execute` on the `Execute Agency Promise As Payment` screen to apply a payment to a promise that has been pre-filled with zeroes, the following exception occurs:

`EvaluationException: Cannot re-add AgencyCyclePayment since it has already been removed.`

Workaround: Use the Payment Wizard to apply the payment instead.

Platform Issues

Keyboard navigation in IE 9 goes to URL bar instead of first editable field (PL-16914)

Issue – If you are using keyboard commands to navigate within Microsoft Internet Explorer 9, IE highlights the URL bar on the first tab, instead of the first editable field.

Workaround – This issue is caused by a change in the IE 9 rendering engine.

Expanding or collapsing a tree node in IE 9 causes screen 'flash' (PL-16919)

Issue – If you are using a tree table within Microsoft Internet Explorer 9 and you expand or collapse a tree node, IE refreshes the screen, causing it to 'flash' momentarily.

Workaround – This issue is caused by a change in the IE 9 rendering engine.

Changing 'Range Radio Button Input' widget in IE 9 causes screen 'flash' (PL-17364)

Issue – If you are using Microsoft Internet Explorer 9 and you change the choice on a Range Radio Button Input screen widget, IE refreshes the screen, causing it to 'flash' momentarily.

Workaround – This issue is caused by a change in the IE 9 rendering engine.

Report folders do not display if empty (PL-1281)

Issue: Guidewire BillingCenter does not display a report folder if that folder is empty.

Workaround: Place an InetSoft replet inside the folder. You can mark the replet as *not visible* within the InetSoft Enterprise Manager. This ensures that the parent folder displays in BillingCenter even if the folder is empty.

Acrobat Sample document template does not allow extra fields (PL-1416)

Issue: Currently, the example Acrobat document template uses two fields, and this requires the same number of fields in `SampleAcrobat.pdf.descriptor` and `SampleAcrobat.pdf` files.

Workaround: To allow more fields in the template and documents created from it, you must update its descriptor file.

Studio PCF editor treats commented-out property values as errors (PL-4582)

Issue: If you attempt to comment out a property value in the Studio PCF editor, Studio treats this as an error.

Workaround: Guidewire is aware of this issue.

Server exception during startup (PL-8167)

Issue: During server startup, there can be a Tomcat server exception. However, after the exception, the server does start up. This is caused by the Tomcat application server attempting to either save or restore sessions using serialization. The session restore causes an exception because BillingCenter sessions contain objects which rely on the metadata being started up.

Workaround: None. The Tomcat application server does start up after the error.

Zone data with missing values cannot be loaded (PL-9123)

Issue: Zone data, which is used for associating postal codes with cities and regions, does not load if it is incomplete.

Workaround: Guidewire is aware of this issue.

H2 development database creates LONGTEXT fields as VARCHAR(65000), instead of as CLOB as in Oracle and SQL Server (PL-9314)

Issue: Guidewire creates schema fields defined as LONGTEXT as LONGVARCHAR columns, instead as CLOB as in the Oracle or SQL Server databases. In addition to the inconsistency involved, this mandates a field length of 65,000 characters or less in LONGTEXT columns in the H2 database.

Workaround: Define a LONGTEXT field as a VARCHAR(...) with the required size, for example, as VARCHAR(120000).

Tab widgets do not work with Microsoft Windows Server 2003 enhanced security enabled (PL-9813)

Issue: If you have the Windows Server 2003 component called IE Enhanced Security Feature installed, tabs may be unresponsive in the user interface.

Workaround: Uninstall this component from the Windows 2003 Server.

Not possible to hot-deploy on all supported application servers (PL-9955)

Issue: It is not possible to hot-deploy on all supported application servers.

Workaround: If you use Tomcat, you are able to see the changes without requiring a restart of the application server. If you use an other application server, you need to perform a restart to see any configuration changes reflected.

First time you click on the arrow of the typekey input, the drop-down menu will not open (PL-10134)

Issue: The drop-down menu does not open on the first click of the arrow on a typekey input. Instead, the help text opens.

Workaround: Turn off help text on focus by setting `InputHelpTextOnFocus` to `false` in the `config.xml` file. By doing that, the help text shows only if you mouse over the input and will not interfere with opening a drop down menu.

InetSoft exception Failed Login Exception: Bad user name anonymous (PL-10148)

Issue: InetSoft generates a Failed Login Exception: Bad user name anonymous exception on server start-up. This occurs if you set the InetSoft logging level to `fine`.

Workaround: This error does not cause any issues. You can safely ignore the exception.

Report still visible after permissions disabled for parent folder (PL-10353)

Issue: An InetSoft report that is included in a permission set is still visible in the report tree even after its parent has been removed from the permission set.

Workaround: You must remove individual reports from permission sets instead of maintaining the reports at the report folder level.

Inability to access typecodes that contain a dash (PL-10532)

Issue: It is not possible to access a typekey code (typecode) that contains a dash using Gosu.

Workaround: Remove any dashes from the typekey codes.

ToolbarButton widget does not render correctly (PL-10742)

Issue: BillingCenter incorrectly renders a `ToolbarButton` widget if the associated `AddMenuItemIterator` is empty.

Workaround: Modify your configuration to disable the menu button if the button contains no menu item.

Must name row iterator element with capital letter (PL-10795)

Issue: The Document search results page (DocumentLV.pcf) throws a `NullPointerException` if the row iterator element name is `document` (with a lower case d).

Workaround: You must use `Document` (with a capital D) instead.

Studio does not properly report verification Error (PL-10822)

Issue: Studio does not properly report a verification error if a typecode contains a trailing space.

Workaround: Remove trailing spaces from all type codes.

BillingCenter displays many warnings when regenerating Javadoc (PL-10829)

Issue: This issue occurs when running the `regen-java-api` and `regen-soap-api` command.

Workaround: Guidewire is aware of this issue.

Navigating between pages in wizard loses data (PL-10920)

Issue: In a wizard pop-ups, it is possible to lose data if you navigate to another page in the wizard.

Workaround: Navigate out of the wizard and then back into it rather than changing steps directly in the wizard.

Double quotes in *Path* environment variable causes problems (PL-10981)

Issue: BillingCenter does not run if there are double quotes in the PATH environment variable.

Workaround: Check the PATH variable and remove any double quotes.

Large number of classes in *plugins\shared\lib* causes studio to fail to start (PL-11026)

Issue: If you attempt to load multiple JAR files that contain many thousands of classes, Studio fails to start and throws an `OutOfMemory` error.

Workaround: Increase `MaxPermSize` to something over `128m` (the default) so that Studio has enough memory to build the type information database.

Ant command *build-war* hangs indefinitely (PL-11045)

Issue: It is possible for the `build-war` Ant command to hang indefinitely. This issue is only reproducible on certain environments.

Workaround: Set the `-noinput` parameter before calling the `build-war` Ant command. For example, instead of `gwbc.bat build-war`, do the following:

```
gwbc.bat -noinput build-war
```

Cache summary statistics rely on server ID being unique (PL-11239)

Issue: If you use the same server ID for multiple servers in a cluster, it can create the following problems:

- Different servers in a cluster deleting or changing each others results
- Deadlocks at the database level

Workaround: By default, the server ID defaults to the host name of the machine running the instance. This works well in many configurations, but causes problems if multiple server instances are running on a single machine. Set the `serverid` system property in the `config.xml` file on the local server host to a unique value in the cluster. If it does not currently exist, then you need to add a `<registry>` element to the `config.xml` file on the local server host.

You can also start the server with the following JVM option: `gw.bc.serverid=????`

See *Defining the Application Server Environment* in the System Administration Guide for details on both of these options.

[Missing display keys cause server to fail to start \(PL-11764\)](#)

Issue: Missing display keys defined in the application default locale can cause the server to fail to start.

Workaround: Within Studio, run the **Verify Resources** utility before attempting to start the application server. BillingCenter bases the type system for display keys on those defined in the default application locale. If a display key in the default application locale is missing and a rule references the missing display key, then BillingCenter cannot compile the rule. Running the resource verification utility can catch these types of missing display keys.

[Countries configured in zone-config.xml still generate a warning during regen-dictionary even if zone data loaded for all of these countries \(PL-11947\)](#)

Issue: Countries configured in `zone-config.xml` still generate a warning during `regen-dictionary` even when zone data is loaded for all of these countries.

Workaround: Warning message is created in error and can safely be ignored.

[The user interface cannot handle starting multiple instances of a batch process \(PL-12372\)](#)

Issue: The user interface cannot handle multiple instances of a batch process.

Workaround: If multiple instances of a batch process need to be executed, they must be started from the command line. Also, you should ensure that the `BatchProcess.isExclusive()` method returns `false` to allow multiple instances to be run simultaneously.

[Debugger break point for the rule engine does not work if first the line is a comment or a blank line \(PL-12912\)](#)

Issue: The debugger break point for the rule engine does not work if the first line is a comment or a blank line.

Workaround: Ensure that the first line is not blank and does not contain a comment. Add comments after the first line.

[Type system refresh after PCF page title change does not update corresponding menu label \(PL-13057\)](#)

Issue: The type system refresh after the PCF page title change does not update corresponding menu label.

Workaround: After updating a page title, the server restart must be done to refresh menu labels and avoid null pointer exceptions due to stale reference.

[Length limitation on entity localization table names \(PL-13360\)](#)

Issue: There is a length limitation on entity localization table names.

Workaround: Ensure that localization tableName property specified in the entity extension file is less than 16 characters. The error message generated if the localization table name exceeds the maximum length indicates that 18 characters are allowed, but that does not account for two additional characters added by the application.

[JavaToolkit.gs has incorrectly hard-coded memory that results in failed regen-java-api Ant task \(PL-13663\)](#)

Issue: File JavaToolkit.gs has hard-coded memory that can result in failed regen-java-api Ant tasks.

Workaround: Increase the size of the max heap setting on line 161 of JavaToolkit.gs in the Ant module. The shipped value is 512.

[US-Locations.txt file with the US geodata from GreatData has special characters that cause validation problems with United States Postal Service \(USPS\) data \(PL-13384\)](#)

Issue: The US-Locations.txt file contains information that does not conform to United States Postal Service (USPS) standards for bulk mailings.

Workaround: The provided US-Locations.txt file is intended only for use in geocoding to identify addresses for a location. You can process the US-Locations.txt file to conform to your particular address standards, and then import that version of the file instead.

[GX models that reference virtual fields and enhancements throw null pointers if null \(PL-13560\)](#)

Issue: The GX models that reference virtual fields and enhancements throw null pointers when they are null.

Workaround: Ensure that null checks and error handling is included so that if referenced virtual fields or enhancements are null, then there will not be a null pointer exception.

[Sending email with file attachment with unicode filename is not correctly handed over to the mail server \(PL-13582\)](#)

Issue: Sending email with file attachment with unicode filename is not sent to the mail server correctly.

Workaround: Use Latin characters for file names on attached files.

[JAXB cannot parse GX model generated XSD \(PL-13598\)](#)

Issue: The GX model generated XSD cannot be parsed by JAXB.

Workaround: Add JAXB annotation elements to the XSD to specify the necessary metadata (such as class names) for JAXB to generate the java class files. Contact Guidewire Support for an example XSD annotated in this way.

[Cannot make a field from a delegate into a localized column \(PL-13761\)](#)

Issue: You cannot make a field from a delegate into a localized column.

Workaround: Move the column that you want to localize off the delegate and onto each of the implementing entities. Then, in order to make the column appear as though it exists on the delegate, define an enhancement property on the delegate, that delegates to the appropriate column, depending on the implementing entity.

[Cannot print second-level list view \(PL-14640\)](#)

Issue: Printing a second-level list view does not work correctly. The print job contains duplicates and does not include all of the items.

Workaround: Remove the lower level list view out of the hierarchical structure and include it at the top level.

RowSet and RowIterator configured in the same ListView do not work correctly. (PL-14946)

Issue: Configuring a RowSet and RowIterator in the same ListView does not provide the right behavior for editability.

Workaround: Separate the RowSet and RowIterator that exist in the same ListView into separate components or into multiple ListView widgets.

Masked input fields do not correctly handle ENTER keypress (PL-14955)

Issue: Input fields that have a field validator mask do not correctly handle the ENTER keypress. After you press ENTER, the field no longer displays the input mask and it is not possible to enter any further input.

Workaround: Cancel the page and re-enter the information.



chapter 31

Guidewire BillingCenter 3.0.5 Release Notes

Release 3.0.5.5

Copyright © 2001-2011 Guidewire Software, Inc. All rights reserved. Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, and the Guidewire logo are registered trademarks, and Guidewire Insurance Suite, Guidewire BillingCenter, Guidewire ContactCenter, and Guidewire Studio are trademarks of Guidewire Software, Inc. All other product names are the trademarks of their respective companies.

This product includes information that is proprietary to Insurance Services Office, Inc (ISO). Where ISO participation is a prerequisite for use of the ISO product, use of the ISO product is limited to those jurisdictions and for those lines of insurance and services for which such customer is licensed by ISO.

This documentation is published as Guidewire Confidential. The contents of this documentation, including product architecture details and APIs, are considered confidential and are fully protected by customer licensing confidentiality agreements and signed Non-Disclosure Agreements (NDAs).

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire BillingCenter.

Version Number

This release of Guidewire BillingCenter is 3.0.5.5.

For Standard Reporting, this release supports InetSoft StyleReport Enterprise Edition 10.1, 20100113.

Support

This document includes important information about the current release of Guidewire BillingCenter. For assistance with this software release, contact Guidewire Customer Support. Contact support on the Web at <http://guidewire.custhelp.com>, by email at support@guidewire.com, or by phone at +1-650-356-4955.

Installing This Release

Refer to the following:

- The *BillingCenter Installation Guide* for general installation information
- The *BillingCenter Reporting Guide* if you plan to use the integrated reporting feature
- BillingCenter release notes previous releases for any versions that have been skipped

InetSoft Evaluation

You can evaluate InetSoft Style Report Enterprise Edition Version 10.1 using the following license key:

L000-72C-ERX-0000BF00100001F-F2B025280327

If you want to use InetSoft and the BillingCenter reporting module in a production environment, contact Guidewire Customer Support to request a full license key. For more information, refer to the *BillingCenter Reporting Guide* documentation.

WARNING The build number of the reporting install files must match the build number of the Guidewire BillingCenter installation. If the build numbers do not match, you will have problems with your reporting installation.

Issues and Major Changes

This section contains issues or changes that can possibly affect your installation:

- Change in Web Service Behavior for Committing Data to the Database (CLM-15982)
- Upgrade Issues

Change in Web Service Behavior for Committing Data to the Database (CLM-15982)

IMPORTANT As part of the change to web services, Guidewire has added a new configuration parameter called `UseSafeBundleForWebServiceOperations`. In the base configuration, Guidewire sets the default value for this parameter to `true`. **If you want to retain the old web services behavior, you must set the parameter value to false.** See the following discussion for details.

The original integration design for Guidewire products focused on ensuring fail-safe integration with external systems and preventing real-time failures due to concurrent data changes. To reduce the chance of unexpected commit errors, during API calls to published web services, the application omitted verification of whether the entity is the most recent version in the database. In other words, the server skipped the check to determine if the entity changed between (a) when the application loaded it from the application cache and (b) when the bundle commits.

This means that you may be at risk to corrupting data in your database if any of the following situations exists in your web services published from a Guidewire product:

- External systems calling web services update data in a parallel manner. If you call the same API concurrently with multiple threads to boost overall throughput, or from multiple integration points, there is risk of corrupting data.
- Web service implementations updating entities that users can modify from the application user interface, or similarly, simultaneous changes from other web service API calls, workflows, batches, or other background processes. Because it is possible for two different threads to modify the same entity, ignoring concurrent data change exception (CDCE) errors can possibly corrupt data. Recent changes can be overwritten.

Impact

Recently, some Guidewire customers have reported issues involving the previously described scenarios. This can manifest as one or more of the following:

- The sequence generator utility returning duplicate (invalid) sequence numbers. When the application commits the bundle, the SQL row insert fails due to non-unique IDs.
- BillingCenter provides a negative or otherwise incorrect value for `TAccount.NumContributingTxns`. This condition is reported by database consistency checks.
- BillingCenter financials consistency checks can fail.
- BillingCenter can experience other data loss issues.

IMPORTANT Published web service APIs that perform only read-only (get) operations are completely unaffected by this issue.

Solution

All Guidewire customers must do the following:

1. Analyze web services usage to see if the above scenarios affect your implementation.
2. Run the database consistency checker tool, also known as `dbcc checker`. This checks for certain types of potentially inconsistent data modified by web services APIs.

IMPORTANT This tool does **not** detect all possible types of corrupted data.

New Web Services Configuration Parameter `UseSafeBundleForWebServiceOperations` (PL-14984)

Guidewire has added a new configuration parameter, `UseSafeBundleForWebServiceOperations`, that changes the behavior of bundle commits in web services published on this server. The default value is `true`.

- If set to `true`, the application detects and does not ignore bean version conflicts.
- If set to `false`, the application ignores bean version conflicts as it commits a bundle.

If you leave this parameter set to `true`, it is possible for a bundle commit to throw a `ConcurrentDataChangeException`. This can happen if another thread or cluster node modified this entity after BillingCenter loaded it from database into the current bundle.

If this error condition occurs, then BillingCenter catches and rethrows the `ConcurrentDataChangeException` as `gw.api.webservice.exception.SOAPRetryableException`. This is a new exception. Guidewire strongly recommends that you update all web service clients to catch this exception and retry the SOAP API call.

Leaving `UseSafeBundleForWebServiceOperations` with its default value of `true` sets the application behavior system-wide. It is possible to reverse this behavior (to make it unsafe) on a method-by-method basis by adding the `@WebServiceIgnoreBeanVersionConflicts` annotation to a method. This annotation only works immediately before a method declaration, not a class declaration, and only on classes with the `@WebService` annotation. There is a known issue that Gosu appears to permit `@WebServiceIgnoreBeanVersionConflicts` on the class level without a compile error. However, it has no effect at the class level.

IMPORTANT It is possible that you previously implemented a workaround for this problem by using the `setIgnoreVersionConflicts` method on the `CommitOptions` at the beginning of SOAP implementation methods. If so, then you must update your client-side logic for detecting and retrying in the case of a concurrent change exception to handle the new exception.

Upgrade Issues

During an upgrade from a previous BillingCenter 3.0.x version, it is possible for the upgrade to fail in the following situations:

- If any line items target the policy period negative writeoff T-account
- If there are any entries in the `CollateralContext` table with a subtype of `CollExcessContext` or `CollPolPerDrawContext`

During the upgrade to BillingCenter 3.0.5, it is possible to encounter the following issues:

- Running `UpgradeCommission` batch process after upgrade throws exception (CC-59387)
- Upgrade of database from BillingCenter 2.1.x to 3.0.5 breaks `UpgradeItemsDBCheckBuilder` consistency check (CC-59340)
- Upgrade of database from BillingCenter 2.1.x to 3.0.5 requires payment plan initialization (CC-59341)

If you encounter any of these issues, please contact Guidewire Customer Support.

Running `UpgradeCommission` batch process after upgrade throws exception (CC-59387)

It is possible for the upgrade process to fail for any `ChargeCommission` that is earned “On Payment Rec’d” and in which one of the following is true:

- The sum of the Commissions Payable balance and Commission Payment transactions is positive **and** its charge's Paid Amount is less than or equal to 0.
- The sum of the Commissions Payable balance and Commission Payment transactions is negative **and** its charge's Paid Amount is greater than or equal to 0.

If the Payable Criteria for any commission earner is “On Payment Rec’d”, then perform the following steps before running the database upgrade utility. The upgrade utility executes the same queries outlined in the following steps and it fails if either query returns any results.

1. Run the Commission Payable batch process. This sets the commissions to a consistent state.
2. Run the two queries listed in **Query 1** and **Query 2**.
3. If either query returns any results, reverse the earning transactions that were caused by the writeoff.

The end result of step 3 is this:

- If the sum of the Commissions Payable balance and Commission Payment transactions is positive, then Charge Paid amount must be positive.

- If the sum of the Commissions Payable balance and Commission Payment transactions is negative, then Charge Paid amount must be negative.

WARNING Do not run the Commission Payable batch process again before the upgrade because it undoes the actions in step 3.

Query 1:

```

SELECT (SELECT ta.BalanceDenorm
FROM bc_taccount ta
INNER JOIN bc_taccountpattern tap
ON ta.TaccountPatternID = tap.ID
WHERE ppcc.hiddentaccountcontainerid = ta.taccountcontainerid
AND tap.TAccountName = 'Commissions Payable') AS Payable,
(SELECT SUM(ii.PaidAmount)
FROM bc_invoiceitem ii
WHERE ii.ChargeID = c.ID) AS PaidAmount,
(SELECT SUM(t.Amount)
FROM bc_transaction t
INNER JOIN bc_producercontext pc
ON pc.TransactionID = t.ID
INNER JOIN bctl_transaction tl
ON t.Subtype = tl.ID
WHERE pc.PolicyProducerCodeChargeID = ppcc.ID
AND tl.TYPECODE = 'PolicyCmsnPayable') AS PayableTxns, ppcc.id
FROM bc_policyproducercodecharge ppcc
INNER JOIN bc_policyproducercode ppc
ON ppcc.policyproducercodeid = ppc.id
INNER JOIN bc_commissionssubplan csp
ON ppc.commissionssubplanid = csp.id
INNER JOIN bctl_payablecriteria crit
ON csp.payablecriteria = crit.id
INNER JOIN bc_taccount ta
ON ppcc.hiddentaccountcontainerid = ta.taccountcontainerid
INNER JOIN bc_taccountpattern tap
ON ta.taccountpatternid = tap.id
INNER JOIN bc_charge c
ON ppcc.chargeid = c.id
WHERE tap.taccountname = 'Commissions Payable'
AND ta.balancedenorm + ISNULL((SELECT SUM(t.amount)
FROM bc_transaction t
INNER JOIN bc_producercontext pc
ON pc.transactionid = t.id
INNER JOIN bctl_transaction tl
ON t.SUBTYPE = tl.id
WHERE pc.policyproducercodechargeid = ppcc.id
AND tl.typecode = 'PolicyCmsnPayable'), 0) < 0
AND crit.typecode = 'paymentreceived'
AND ISNULL((SELECT SUM(ii.paidamount)
FROM bc_invoiceitem ii
WHERE ii.chargeid = c.id), 0) >= 0

```

Query 2:

```

SELECT (SELECT ta.BalanceDenorm
FROM bc_taccount ta
INNER JOIN bc_taccountpattern tap
ON ta.TaccountPatternID = tap.ID
WHERE ppcc.hiddentaccountcontainerid = ta.taccountcontainerid
AND tap.TAccountName = 'Commissions Payable') AS Payable,
(SELECT SUM(ii.PaidAmount)
FROM bc_invoiceitem ii
WHERE ii.ChargeID = c.ID) AS PaidAmount,
(SELECT SUM(t.Amount)
FROM bc_transaction t
INNER JOIN bc_producercontext pc
ON pc.TransactionID = t.ID
INNER JOIN bctl_transaction tl
ON t.Subtype = tl.ID
WHERE pc.PolicyProducerCodeChargeID = ppcc.ID
AND tl.TYPECODE = 'PolicyCmsnPayable') AS PayableTxns, ppcc.id
FROM bc_policyproducercodecharge ppcc
INNER JOIN bc_policyproducercode ppc
ON ppcc.policyproducercodeid = ppc.id
INNER JOIN bc_commissionssubplan csp
ON ppc.commissionssubplanid = csp.id
INNER JOIN bctl_payablecriteria crit
ON csp.payablecriteria = crit.id

```

```

    INNER JOIN bc_taccount ta
    ON ppcc.hiddentaccountcontainerid = ta.taccountcontainerid
    INNER JOIN bc_taccountpattern tap
    ON ta.taccountpatternid = tap.id
    INNER JOIN bc_charge c
    ON ppcc.chargeid = c.id
    WHERE tap.taccountname = 'Commissions Payable'
    AND ta.balancedenorm + ISNULL((SELECT SUM(t.amount)
        FROM bc_transaction t
        INNER JOIN bc_producercontext pc
        ON pc.transactionid = t.id
        INNER JOIN bctl_transaction tl
        ON t.SUBTYPE = tl.id
        WHERE pc.policyproducercodechargeid = ppcc.id
        AND tl.typecode = 'PolicyCmsnPayable'), 0) > 0
    AND crit.typecode = 'paymentreceived'
    AND ISNULL((SELECT SUM(ii.paidamount)
        FROM bc_invoiceitem ii
        WHERE ii.chargeid = c.id), 0) <= 0

```

Upgrade of database from BillingCenter 2.1.x to 3.0.5 breaks *UpgradeItemsDBCheckBuilder* consistency check (CC-59340)

Issue: Upgrading the BillingCenter database from 2.1.x to 3.0.5 breaks the *UpgradeItemsDBCheckBuilder* consistency check. If this happens, you see the following error message:

One or more Invoice Items found with type 'Upgrade Commission'.
Please run the UpgradeCommission Work Queue.

Workaround: After upgrading the database from BillingCenter 2.1.x to 3.0.5, run the following batch process:

`UpgradeCommission`

The process determines how to distribute direct bill and non-primary agency bill commission from the "Upgrade Commission" invoice items to the item level on the actual invoice items. It then removes all "Upgrade Commission" invoice items. You need to run `UpgradeCommission` a single time only after the upgrade triggers have been run.

Upgrade of database from BillingCenter 2.1.x to 3.0.5 requires payment plan initialization (CC-59341)

Issue: Upgrading the BillingCenter database from 2.1.x to 3.0.5 requires that you initialize the payment plan field of policy billing instructions.

Workaround: After upgrading the database from BillingCenter C 2.1.x. to 3.0.5, run the following batch process:

`UpgradeBillingInstructionPaymentPlan`

The process initializes the payment plan field of policy billing instructions.

Changes in this Release

This section describes the product changes in this release.

- Base PCF File Changes
- Improvements to Upgrade Diff Report
- Improvements and General Issues

Base PCF File Changes

All links below require the `ReleaseNotes_files` directory on your local disk.

BillingCenter release 3.0.4 to 3.0.5

- To view a report of the changes in the base PCF files in the `modules/bc` directory, [click here](#).
- To view a report of the changes in the base PCF files in the `modules/p1` directory, [click here](#).

Improvements to Upgrade Diff Report

With previous releases, Guidewire provided a static report in the release notes detailing certain differences between the current release and the prior minor release. This report described changes in display keys, entities, typelists, and the Gosu API. Guidewire has improved this report to dynamically conform to each customer's particular upgrade path. In addition, the report is regularly updated as the tools for generating it are enhanced to provide more information.

Because the new report is tailored to your particular upgrade requirements, it is no longer included with the general release notes. To obtain your custom Upgrade Diff Report, contact your Guidewire representative. In the future, this report will be available on the Guidewire Resource Center web portal, allowing you to view the latest version right when you need it.

Improvements and General Issues

The following are the improvements added to and the general issues corrected in this release:

ID	Description
Account	
CC-59480	Fixed an issue where the <code>prodedit</code> permission was being checked when a payment was applied to an account. Now a user without <code>prodedit</code> permission can execute a payment using the New Direct Bill Payment screen.
Agency Bill	
CC-59308	Fixed an issue with purging completed workflows from the database. Previously, if the server was reloaded after the user ran the Purge Workflow and Purge Workflow Logs batch processes and then cleared global cache, a null pointer exception was thrown when a user accessed the Agency Bill Cycles screen.
CC-59350	To improve performance and decrease the number of unnecessary rows in the database, a distribution item is no longer created for an invoice item with no payment applied to it ("zero dollar dist item"). This change also means that agency bill Payment and Transaction screens are cleaner and less cluttered. All behavior with respect to exception handling is unchanged. There are four circumstances under which a zero dollar distribution item is committed to the database: <ul style="list-style-type: none">• If there is a comment associated with the distribution item• If the distribution item modifies a distribution item that has a payment applied to it• If the distribution item is for a commission-only invoice item• If the distribution item has a disposition of <code>writeoff</code> This improvement has caused the following change in the behavior of the Agency Payment wizard. When you use the Agency Payment wizard to apply a payment to a promise that has been pre-filled with zeroes, the zero-filled items are not listed in the Policy Activity list view in step 1 of the wizard. You need to click Add Items to add any zero-filled items that you wish to pay. All behavior with respect to exception handling is unchanged.
CC-59469	BillingCenter no longer snapshots invoice items that are placed on planned statement invoices. This change makes snapshotting of agency bill invoice items more performant. It does not affect functional behavior seen by end-users. The throughput of the Statement Bill batch process has improved up to 80% in Internal tests. The number of configured worker instances may need to be adjusted to achieve optimal performance.
CC-59488	Fixed a problem where agency suspense promise items would not show up on the My Agency Items screen if a payment against a promise was reversed.
CC-59490	Improved the performance of snapshotting agency bill snapshot invoice items. This improvement does not affect functional behavior seen by end-users. The number of configured worker instances may need to be adjusted to achieve optimal performance.
Charge/Invoice	

ID	Description
CC-59117	<p>Guidewire has modified the <i>Invoice Due</i> batch process to use a similar logic as the <i>Invoice</i> batch process. As a consequence, Guidewire has added a new configuration parameter—<i>InvoicePeriodDueCutoff</i>—that takes a time value as part of this modification. The <i>Invoice Due</i> batch process subtracts the time from the current time. Any invoice with a due date prior to that time is considered for processing.</p> <p>The code logic looks similar to the following:</p> <pre>select * from bc_invoice i where i.PaymentDueDate <= current_time - InvoicePeriodDueCutoff</pre> <p>The default for <i>InvoicePeriodDueCutoff</i> is 12:00 a.m. to ensure that, without configuration, the batch process behaves the same as in prior versions.</p>
Commission/Commission Plan	
CC-59301	Fixed an issue where overriding the commission rate at charge level after the payer of the charge was assigned to a producer would fail with an <i>EvaluationException</i> . Now, when you reassign the payer of a charge or an invoice item from an account to a producer or a producer to another producer and subsequently override the commission rate, the override succeeds.
Core	
PL-14410	<p>Modified how BillingCenter handles proximity searching:</p> <ul style="list-style-type: none"> Added a <i>RadiusSearchMaxResults</i> property to the <i>ProximitySearchParameters</i> non-persistent entity. This property indicates the maximum number of items to return on a radius search. BillingCenter ignores this value if the value of <i>DistanceBasedSearch</i> is <i>false</i>. If the value of <i>DistanceBasedSearch</i> is <i>true</i>, and the value of <i>RadiusSearchMaxResults</i> is <i>null</i>, zero, or negative, then BillingCenter limits the search by the value of <i>ProximityRadiusSearchDefaultMaxResultCount</i>. Added configuration parameter <i>ProximityRadiusSearchDefaultMaxResultCount</i> with a default of 1000. This parameter sets the maximum number of results that a proximity search is to return if using a fixed radius (n miles/km). BillingCenter ignores this parameter if performing an ordinal (nearest-n) search.
PL-14478	Localization tables for localized columns on effective-dated entities were not properly included in the effective-dated graph. Essentially, these tables needed to become effective-dated as well. This change makes this happen.
PL-15728	<p>Added a new optional attribute, <i>forceEqMatchType</i>, that can be added to the <i>Criterion</i> element in a search configuration file. If this attribute is present, it must be the name of a Boolean property on the criteria entity.</p> <ul style="list-style-type: none"> If this attribute evaluates to <i>true</i>, the <i>Criterion</i> uses an <i>eq</i> (equality) match. If this attribute evaluates to <i>false</i>, the <i>Criterion</i> uses the <i>matchType</i> specified in the <i>Criterion</i>. <p>For example:</p> <pre><Criterion property="StringProperty" forceEqMatchType="FlagProperty" matchType="startsWith"/></pre> <p>This <i>Criterion</i> uses a <i>startsWith</i> match for <i>StringProperty</i> <i>unless</i> the <i>FlagProperty</i> on the criteria entity is <i>true</i>, in which case the <i>Criterion</i> uses an <i>eq</i> match type.</p>
PL-15958	Fixed an issue in which stopping the global cache did not stop any currently running background thread(s) and actually started them if they were not already started.
PL-15959	<p>Corrected an issue with a null pointer in a global cache reaper thread that prevented the BillingCenter application server running on WebSphere from shutting down gracefully. The issue caused the following error to occur:</p> <pre>ERROR Unexpected error while in global cache reaper thread: java.lang.RuntimeException: java.lang.NullPointerException</pre>
Database	
PL-12073	Added the ability to override the <i>createhistogram</i> attribute on the <i>CreateTime</i> column for a specific table. If you want to force the creation of a histogram on an base configuration column, then you can do so by adding a <i><histogramstatistics></i> element for the column in the appropriate <i><tablestatistics></i> element of the <i><databasestatistics></i> element in <i>config.xml</i> .

ID	Description
PL-12200	<p>Modified the behavior of (and slightly renamed) the DBMS Counter Threshold (millis) option on the (Server Tools) Guidewire Profiler configuration page:</p> <ul style="list-style-type: none"> Name change - Renamed DBMS Counter Threshold (millis) to DBMS Instrumentation Capture Threshold for each Action (millis). Behavior change - Formerly, this option generated a DBMS report if any database operation exceeded a preset threshold value. Now, the profiler generates a DBMS report if the action (start of profiling session, end of profiling session) exceeds the threshold value set by this option. New option parameter Diff DBMS Counters - Set to capture the DBMS counters at the beginning of the profiling session and to include analysis of the differences in the DBMS-specific report. This option is meaningful only if you set the first option.
PL-15314	<p>Enhanced the consistency check failure message. Previously, the failure message from a consistency check violation listed the SQL query that was executed and a description of the problem. However, the SQL query did not return the actual rows that caused the violation. Guidewire has added the ability for you to include a separate SQL query that will identify the specific rows, along with any other pertinent information.</p>
PL-15867	<p>Corrected an issue that occurred while attempting to download database statistics if the database contained a column that was not in the base application data model and there was a histogram on the column. BillingCenter now logs a warning and permits the download to continue.</p>
PL-15873	<p>Guidewire now persists the results of consistency checks to the database for use of performance monitoring. You can monitor the progress of a particular consistency check run by using SQL queries against the associated tables, which are:</p> <ul style="list-style-type: none"> xx_dbconsistcheckrun xx_dbconsistcheckqueryexec <p>The prefix, xx, is the application prefix.</p> <p>Entity DBConsistCheckRun represents a run of the consistency checker. It contains an array of DBConsistCheckQueryExec entities, which represent the execution of an individual consistency check within a run.</p>
Integration	
PL-14412	<p>Fixed an issue in which BillingCenter did not populate the description field properly as it created a document from a template if the description existed in the template descriptor.</p>
PL-14960	<p>BillingCenter now handles the encoding attribute on the XML header (the processing instruction). This attribute indicates the character encoding of the XML document. For example:</p> <pre><?xml version="1.0" encoding="UTF-8"?></pre> <p>For a discussion of XML character encoding, see the following:</p> <p>http://www.w3.org/International/0-charset</p> <p>Note: Guidewire supports the Java UTF-16BE and UTF-16LE character sets as UTF-16 XML encoding, with the appropriate handling for byte-order bytes.</p>
PL-14984	<p>Added configuration parameter UseSafeBundleForWebServiceOperations. The default value in the base configuration is true.</p> <p>See Change in Web Service Behavior for Committing Data to the Database (CLM-15982) for details.</p>
PL-15327	<p>Added configuration parameter ReCallDocumentContentSourceAfterRollback. This parameter causes BillingCenter to call the isDocument and addDocument methods again if there is a roll-back during a bundle commit.</p> <p>If a roll-back of the bundle does occur, BillingCenter rolls back each entity in the bundle. The initial commit extracted the document content from the web request. The document content no longer exists after the bundle commit. Therefore, it is not possible to roll-back the document ID, even though it is possible to retrieve all other document fields. This can cause problems to occur.</p> <p>Guidewire recommends the following:</p> <ul style="list-style-type: none"> If you have extended the Document entity, then set this parameter to true. If you have not extended the Document entity, then it is most likely that you want to set this parameter to false. If you are using a IDocumentMetadataSource, then set the parameter to false. If you then experience validation exceptions, test your set up, and, if necessary, set this value to true.
Miscellaneous	

ID	Description
CC-59591	A consistency check was implemented to look for DirectBillPayment objects in the database that are not reversed and do not have a DistributedDate. This ensures that direct bill payments are not committed unless they have been executed.
CC-59628	Modified the New Credit wizard to make the Credit Type field required.
CC-59654	Fixed an issue with the Agency Distribution Disposition plugin, where the special handling logic created writeoffs with incorrect gross and commission writeoff amounts. The plugin now calculates the gross and commission writeoff amounts correctly.
Performance	
CC-59330	Improved the performance of consistency checks, particularly on Oracle databases. Previously, consistency checks on databases with more than 100,000 accounts could take several hours.
CC-59454	Running the CommissionPayable batch process now results in less database disk IO. Internal tests show a 170% increase in PolicyCommissions processed per minute. The number of configured worker instances may need to be adjusted to achieve optimal performance.
CC-59459	Improved performance of the InvoiceBill batch process, which is now less susceptible to ConcurrentDataChangeException exceptions. In internal tests, throughput was improved by 20%. The number of configured worker instances may need to be adjusted to achieve optimal performance.
CC-59537	Improved the performance of the staging table load by rewriting several SQL statements.
Reinstatement Process	
CC-58728	Previously, when a policy was flat cancelled, and then reinstituted , the policy closure status was not changed from CLOSED to OPEN. Now, after a flat-cancelled policy is reinstated, the policy closure status is set to OPEN or OPEN_LOCKED as appropriate.
Rewrite	
CC-59629	Before this fix, when a policy was flat cancelled and then rewritten , the policy closure status was not changed from CLOSED to OPEN. Now, after a flat cancellation followed by a rewrite, the policy closure status is set to OPEN or OPEN_LOCKED as appropriate.
Trouble Ticket	
CC-56402	Previously, two methods used to create trouble tickets appeared in Studio as deprecated: <ul style="list-style-type: none"> • TroubleTicket.addToTroubleTicketJoinEntities • Account.addToTroubleTicketJoinEntities Studio no longer flags these methods as deprecated.
CC-59300	Fixed an issue where the Release Trouble Ticket Hold Types batch process failed to initiate delinquency workflow for a trouble ticket that was no longer held. Now, after the batch process releases the hold on a trouble ticket whose release date has passed, it then creates the associated delinquency workflow.
Studio	
PL-10504	Corrected a Studio issue in which debugging in Browser mode resulted in an out-of-memory error.
PL-10553	Corrected an issue in which debugging a rule in Studio did not display and highlight the current line number in the source window.
PL-13282	Corrected a display issue in which Studio truncated the error message that it showed if you hovered the mouse over a non-valid widget in the PCF editor.
PL-14174	Corrected an issue with the Studio auto-save feature that did not work correctly if you used CTRL-N to open another view before saving the view that was currently open.
PL-14367	Added a localization file import command to the command line system tools with the following syntax: <pre>import-l10ns -Dimport.file=<translation file> -Dimport.locale=<destination locale></pre> This command provides the same functionality as that of the Studio Localizations → Display Keys (or Typelist Localization) right-click menu command Import Translation File.
PL-14518	Corrected an issue with the Studio rule debugger in which stepping through breakpoints set in rules did not update the Debugger pane (at the bottom of the screen) properly.
PL-15430	Modified Studio behavior so that an incorrect typelist or entity definition does not prevent Studio from starting. Instead, Studio logs the error in the Studio console. It is also possible to correct the error without requiring a restart of Studio.
Web	

ID	Description
PL-13053	Modified the way that BillingCenter handles validation in the user interface. Checking, then unchecking an input group and then moving to another wizard step no longer triggers validation. Guidewire recommends that you do not rely on this behavior as there are other validations that BillingCenter does not execute either (for example, the required property).
PL-13792	Corrected an interface issue in which copying and pasting a phone number that included an extension did not paste correctly into another field.
PL-15246	Corrected a button display problem with Internet Explorer 7 and Internet Explorer 8 that occurred if all of the following were true: <ul style="list-style-type: none"> • The button was disabled. • The header cell was blank or header text was shorter than the button.

Known Issues and Limitations

Note: For maintenance releases, it is Guidewire policy to defer fixing configuration issues if they require merging files during the upgrade. Workarounds to many of these issues are listed in the following sections. The goal of this policy is to make upgrades as straightforward as possible.

This section describes known issues with this release of Guidewire BillingCenter:

- BillingCenter Issues
- Platform Issues

BillingCenter Issues

On AIX without an X11 server, BillingCenter may report that an X11 resource is missing (CC-5889)

Issue: BillingCenter does not require an X11 server; however other components of the BillingCenter environment (such as Java) may inadvertently have this requirement. If so, BillingCenter may report that an X11 resource is missing when it is starting up.

Workaround: Install an X11 server, or contact Guidewire support to determine if the dependency can be removed.

Incorrect additivity statements in the default *logging.properties* file (CC-50615)

Issue: Some additivity statements in the *logging.properties* file are incorrect and therefore, ineffective. An additivity statement attempts to keep the category from sending its log messages up to its parent, preventing duplicate log messages across files.

Workaround: In the *logging.properties* file (located at *modules/cc/config/logging*), change:

```
log4j.additivity.PluginsLog=false
to
log4j.additivity.Plugin=false
```

Problems occurred with the use of negative collateral charges (CC-51311)

Issue: When issuing a negative collateral charge, the amount gets subtracted from a Charge Due T-account and a payer's Unapplied T-account. This causes an imbalance between *collateral_reserve* and *collateral_held* when issuing a negative charge.

Workaround: Guidewire is aware of this issue.

Automatic distribution to policies not working correctly (CC-51511)

Issue: Automatic distribution to policies on Account → Disbursement screen does not distribute the payment correctly.

Workaround: Guidewire is aware of this issue.

Billing instructions for Issuance and Policy Change must work with premium and other charges (Premium Reporting Only) (CC-52032)

Issue: This issue is for Premium Reporting policy periods only. Users are note allowed to enter premium charges as part of an issuance or policy change and process them as any other policy period. This issue applies to all billing instructions types.

Workaround: Guidewire is aware of this issue.

No commission reserve and onset transfer transaction was created when a direct bill item was reassigned to an agency bill producer (CC-54923)

Issue: When a direct bill policy has no specified primary producer and one of the policy's item is reassigned to an agency bill producer, no commission reserve and onset transfer transaction created. This was because there is no initial reserve.

Workaround: Guidewire is aware of this issue.

New collateral charges and invoice amounts are being generated based on previously paid collateral amounts, not the previously billed or planned amounts (CC-55209)

Issue: New collateral charges and invoice amounts are being generated based on the amount of collateral previously paid and do not account for amounts of collateral previously Billed or Planned (not yet billed). This results in an over billing of the collateral amount on the invoices.

Workaround: Guidewire is aware of this issue.

Errors occur in the totals for the producer's rate on the Transactions screen (CC-55835)

Issue: A difference occurs in the totals shown on the Transactions screen for a producer when selecting Reserves and then choosing either the Last 30 Days or Show All filter.

Workaround: Guidewire is aware of this issue.

Account Evaluation is not counting account level delinquencies (CC-55938)

Issue: The count for number of delinquencies on AccountEvaluation screen doesn't factor account level delinquencies.

Workaround: Use the Gosu finder to identify the account level delinquency processes and add it to the BillingCenter value.

After a reversal, the distributed payment is still being reserved for the premium payment (CC-56538)

Issue: The disbursement amount is not fully moved to disbursement and does not show up on the Disbursement page although an entire payment has been reversed and the associated charge had been assigned to another account.

Workaround: This is a new feature that is not fully implemented. Guidewire is aware of this issue.

Moving a Billed invoice item should negate the direct bill commission paid, not the agency bill commission paid (CC-56656)

Issue: If an item is moved from a Billed invoice to a Planned invoice, an earnings transaction is created for the Payable amount on the commission. However, if the commission was earned and paid, it is not being negated.

Workaround: Guidewire is aware of this issue.

Incorrect totals in the policy payment schedule occurred on agency bill statements after a change was made to one of two producers on a policy (CC-56925)

Issue: The producer statement does not show the correct dollar amount after a change is made to one of the policy's payment plans on a policy with two producers after the statement was billed and partially paid.

Workaround: Guidewire is aware of this issue.

Should not be able to edit Payment Plan after it has been used by a policy (CC-57758)

Issue: In the base configuration, it is possible to edit a Payment Plan even after it has been used by a policy. This is incorrect. Only the expiration date should be updateable.

Workaround: Configure the payment plan screen to only allow the expiration date to be updated.

Invoice PreUpdate rule is bypassed when a new invoice created for Collateral charge (CC-58254)

Issue: Automatic distribution to policies on Account → Disbursement screen does not distribute the payment correctly.

Workaround: Guidewire is aware of this issue.

No API to remove invoice items from DirectBillPaymentView (CC-58397)

Issue: There is no way to programmatically remove an Invoice item (or invoice item array) from a distribution.

Workaround: Guidewire is aware of this issue.

Use of StringBuilder Class prevents debugging of web service (CC-58533)

Issue: In a web service, if you have a method with a parameter of type X and inside the web service method you reference a Gosu class of type Y. And, Gosu class Y has a `toString()` method (overriding the default) that uses a `java.lang.StringBuilder` class. If you set a breakpoint in the web service method and try to step through the method, then you will get errors.

Workaround: Guidewire is aware of this issue.

Reversing Write-Off Commissions and Write-Off Commission Recovery in the UI (CC-58655)

Issue: The BillingCenter user interface does not provide a way reverse Write-Off Commissions and Write-Off Commission Recovery.

Workaround: Clone the Reverse button under the Payments tab and reverse `CmsnPayableReduction` and other transactions using the following action: `CurrentLocation.startEditing(); producerTransaction.reverse();` and `CurrentLocation.commit();`.

Cannot un-distribute direct bill payment (CC-58764, CC-58764)

Issue: The base configuration does not allow a direct bill payment to be undistributed.

Workaround: Use the configuration to create an **Unapply** button on the **Payment List View**. This button should remove the original **DirectBillMoneyRcvd**, reverse all associated transactions, and then create a brand new **DBMoneyRcvd** with the same attributes as the original.

List view on New Account and New Producer wizard does not display properly (CC-59574)

Issue: Attempting to add an existing contact following a new contact causes the added contact to not display properly in the list view on the **Add Contact** wizard on the **New Account** and **New Producer** screens.

Workaround: Click on a column heading and the sort cause the screen to refresh and display the contacts correctly.

Agency Bill Payment wizard generates *EvaluationException* (CC-59721)

Issue: Currently, if you click **Execute** on the **Execute Agency Promise As Payment** screen to apply a payment to a promise that has been pre-filled with zeroes, the following exception occurs:

EvaluationException: Cannot re-add AgencyCyclePayment since it has already been removed.

Workaround: Use the **Payment Wizard** to apply the payment instead.

NetAmountDistributed on *BaseDist* not calculated correctly for modifying payment with suspense items (PL-59776)

Issue: When a user edits a direct bill or agency bill payment that has suspense items in it, the net amount distributed field on the payment may be calculated incorrectly. This error occurs when the payment being edited already has one or more suspense items in it, and the user modifies only the payment item amounts and not the suspense item amounts.

Workaround: Guidewire is aware of this issue.

Platform Issues

Report folders do not display if empty (PL-1281)

Issue: Guidewire BillingCenter does not display a report folder if that folder is empty.

Workaround: Place an InetSoft replet inside the folder. You can mark the replet as *not visible* within the InetSoft Enterprise Manager. This ensures that the parent folder displays in BillingCenter even if the folder is empty.

Acrobat Sample document template does not allow extra fields (PL-1416)

Issue: Currently, the example Acrobat document template uses two fields, and this requires the same number of fields in **SampleAcrobat.pdf.descriptor** and **SampleAcrobat.pdf** files.

Workaround: To allow more fields in the template and documents created from it, you must update its descriptor file.

Application startup fails with “Exception committing consistency check instrumentation” error (PL-16208)

Issue: This error occurs if you set the following `<database>` element in `config.xml` and then attempt to start the application server:

`checker="true"`

Workaround: Set this value to `false` before starting the application server.

[Studio PCF editor treats commented-out property values as errors \(PL-4582\)](#)

Issue: If you attempt to comment out a property value in the Studio PCF editor, Studio treats this as an error.

Workaround: Guidewire is aware of this issue.

[Server exception during startup \(PL-8167\)](#)

Issue: During server startup, there can be a Tomcat server exception. However, after the exception, the server does start up. This is caused by the Tomcat application server attempting to either save or restore sessions using serialization. The session restore causes an exception because BillingCenter sessions contain objects which rely on the metadata being started up.

Workaround: None. The Tomcat application server does start up after the error.

[Zone data with missing values cannot be loaded \(PL-9123\)](#)

Issue: Zone data, which is used for associating postal codes with cities and regions, does not load if it is incomplete.

Workaround: Guidewire is aware of this issue.

[H2 development database creates LONGTEXT fields as VARCHAR\(65000\), instead of as CLOB as in Oracle and SQL Server \(PL-9314\)](#)

Issue: Guidewire creates schema fields defined as LONGTEXT as LONGVARCHAR columns, instead as CLOB as in the Oracle or SQL Server databases. In addition to the inconsistency involved, this mandates a field length of 65,000 characters or less in LONGTEXT columns in the H2 database.

Workaround: Define a LONGTEXT field as a VARCHAR(...) with the required size, for example, as VARCHAR(120000).

[Tab widgets do not work with Microsoft Windows Server 2003 enhanced security enabled \(PL-9813\)](#)

Issue: If you have the Windows Server 2003 component called IE Enhanced Security Feature installed, tabs may be unresponsive in the user interface.

Workaround: Uninstall this component from the Windows 2003 Server.

[Not possible to hot-deploy on all supported application servers \(PL-9955\)](#)

Issue: It is not possible to hot-deploy on all supported application servers.

Workaround: If you use Tomcat, you are able to see the changes without requiring a restart of the application server. If you use an other application server, you need to perform a restart to see any configuration changes reflected.

[First time you click on the arrow of the typekey input, the drop-down menu will not open \(PL-10134\)](#)

Issue: The drop-down menu does not open on the first click of the arrow on a typekey input. Instead, the help text opens.

Workaround: Turn off help text on focus by setting InputHelpTextOnFocus to false in the config.xml file. By doing that, the help text shows only if you mouse over the input and will not interfere with opening a drop down menu.

InetSoft exception Failed Login Exception: Bad user name anonymous (PL-10148)

Issue: InetSoft generates a Failed Login Exception: Bad user name anonymous exception on server start-up. This occurs if you set the InetSoft logging level to *fine*.

Workaround: This error does not cause any issues. You can safely ignore the exception.

Report still visible after permissions disabled for parent folder (PL-10353)

Issue: An InetSoft report that is included in a permission set is still visible in the report tree even after its parent has been removed from the permission set.

Workaround: You must remove individual reports from permission sets instead of maintaining the reports at the report folder level.

Inability to access typecodes that contain a dash (PL-10532)

Issue: It is not possible to access a typekey code (typecode) that contains a dash using Gosu.

Workaround: Remove any dashes from the typekey codes.

ToolbarButton widget does not render correctly (PL-10742)

Issue: BillingCenter incorrectly renders a ToolbarButton widget if the associated AddMenuItemIterator is empty.

Workaround: Modify your configuration to disable the menu button if the button contains no menu item.

Must name row iterator element with capital letter (PL-10795)

Issue: The Document search results page (DocumentLV.pcf) throws a NullPointerException if the row iterator element name is *document* (with a lower case d).

Workaround: You must use *Document* (with a capital D) instead.

Studio does not properly report verification Error (PL-10822)

Issue: Studio does not properly report a verification error if a typecode contains a trailing space.

Workaround: Remove trailing spaces from all type codes.

BillingCenter displays many warnings when regenerating Javadoc (PL-10829)

Issue: This issue occurs when running the regen-java-api and regen-soap-api command.

Workaround: Guidewire is aware of this issue.

Navigating between pages in wizard loses data (PL-10920)

Issue: In a wizard pop-ups, it is possible to lose data if you navigate to another page in the wizard.

Workaround: Navigate out of the wizard and then back into it rather than changing steps directly in the wizard.

Values in MIME-Type drop-down not localized (PL-10921)

Issue: BillingCenter does not localize the values in the MIME type drop-down list properly.

Workaround: If you want to localize the MIME type names, you can do so by modifying the *description* attribute on the <mimetype> element definitions in config.xml. Currently, BillingCenter does not support MIME type definitions in multiple languages.

Studio—with Subversion (SVN)—incorrectly copies .svn files (PL-10932)

Issue: This issue effects installations that use Subversion only. If you modify the base configuration rules in Studio, Studio creates a copy of the rule in the modules/configuration folder. Studio then executes Subversion commands to add the files and folders to Subversion. In the process, Studio incorrectly copies the .svn folder and its children. This leads to Subversion problems.

Workaround: Manually delete the copied .svn folders

Double quotes in Path environment variable causes problems (PL-10981)

Issue: BillingCenter does not run if there are double quotes in the PATH environment variable.

Workaround: Check the PATH variable and remove any double quotes.

Large number of classes in *plugins\shared\lib* causes studio to fail to start (PL-11026)

Issue: If you attempt to load multiple JAR files that contain many thousands of classes, Studio fails to start and throws an OutOfMemory error.

Workaround: Increase MaxPermSize to something over 128m (the default) so that Studio has enough memory to build the type information database.

Ant command *build-war* hangs indefinitely (PL-11045)

Issue: It is possible for the *build-war* Ant command to hang indefinitely. This issue is only reproducible on certain environments.

Workaround: Set the -noinput parameter before calling the *build-war* Ant command. For example, instead of gwcc.bat build-war, do the following:

```
gwbc.bat -noinput build-war
```

Cache summary statistics rely on server ID being unique (PL-11239)

Issue: If you use the same server ID for multiple servers in a cluster, it can create the following problems:

- Different servers in a cluster deleting or changing each others results
- Deadlocks at the database level

Workaround: By default, the server ID defaults to the host name of the machine running the instance. This works well in many configurations, but causes problems if multiple server instances are running on a single machine. Set the `serverid` system property in the `config.xml` file on the local server host to a unique value in the cluster. If it does not currently exist, then you need to add a `<registry>` element to the `config.xml` file on the local server host.

You can also start the server with the following JVM option: `gw.bc.serverid=????`

See *Defining the Application Server Environment* in the System Administration Guide for details on both of these options.

Missing display keys cause server to fail to start (PL-11764)

Issue: Missing display keys defined in the application default locale can cause the server to fail to start.

Workaround: Within Studio, run the Verify Resources utility before attempting to start the application server. BillingCenter bases the type system for display keys on those defined in the default application locale. If a display key in the default application locale is missing and a rule references the missing display key, then BillingCenter cannot compile the rule. Running the resource verification utility can catch these types of missing display keys.

Countries configured in `zone-config.xml` still generate a warning during `regen-dictionary` even if zone data loaded for all of these countries (PL-11947)

Issue: Countries configured in `zone-config.xml` still generate a warning during `regen-dictionary` even when zone data is loaded for all of these countries.

Workaround: Warning message is created in error and can safely be ignored.

The user interface cannot handle starting multiple instances of a batch process (PL-12372)

Issue: The user interface cannot handle multiple instances of a batch process.

Workaround: If multiple instances of a batch process need to be executed, they must be started from the command line. Also, you should ensure that the `BatchProcess.isExclusive()` method returns `false` to allow multiple instances to be run simultaneously.

Debugger break point for the rule engine does not work if first the line is a comment or a blank line (PL-12912)

Issue: The debugger break point for the rule engine does not work if the first line is a comment or a blank line.

Workaround: Ensure that the first line is not blank and does not contain a comment. Add comments after the first line.

Type system refresh after PCF page title change does not update corresponding menu label (PL-13057)

Issue: The type system refresh after the PCF page title change does not update corresponding menu label.

Workaround: After updating a page title, the server restart must be done to refresh menu labels and avoid null pointer exceptions due to stale reference.

Length limitation on entity localization table names (PL-13360)

Issue: There is a length limitation on entity localization table names.

Workaround: Ensure that localization `tableName` property specified in the entity extension file is less than 16 characters. The error message generated if the localization table name exceeds the maximum length indicates that 18 characters are allowed, but that does not account for two additional characters added by the application.

Use of `StringBuilder` class prevents debugging of web service (PL-13435)

Issue: Suppose that you have a web service in which you have a method with a parameter of type X and inside the web service method you reference a Gosu class of type Y. This can be an issue if the Gosu class Y has a `toString` method (overriding the default) that uses a `java.lang.StringBuilder` class. If you attempt to set a breakpoint in the web service method and try to step through the method, Studio generates errors.

Workaround: Studio does not generate errors if you call the web service in normal run mode rather than in debug mode.

`JavaToolkit.gs` has incorrectly hard-coded memory that results in failed `regen-java-api` Ant task (PL-13663)

Issue: File `JavaToolkit.gs` has hard-coded memory that can result in failed `regen-java-api` Ant tasks.

Workaround: Increase the size of the max heap setting on line 161 of `JavaToolkit.gs` in the Ant module. The shipped value is 512.

US-Locations.txt file with the US geodata from GreatData has special characters that cause validation problems with United States Postal Service (USPS) data (PL-13384)

Issue: The US-Locations.txt file contains information that does not conform to United States Postal Service (USPS) standards for bulk mailings.

Workaround: The provided US-Locations.txt file is intended only for use in geocoding to identify addresses for a location. You can process the US-Locations.txt file to conform to your particular address standards, and then import that version of the file instead.

GX models that reference virtual fields and enhancements throw null pointers if null (PL-13560)

Issue: The GX models that reference virtual fields and enhancements throw null pointers when they are null.

Workaround: Ensure that null checks and error handling is included so that if referenced virtual fields or enhancements are null, then there will not be a null pointer exception.

Sending email with file attachment with unicode filename is not correctly handed over to the mail server (PL-13582)

Issue: Sending email with file attachment with unicode filename is not sent to the mail server correctly.

Workaround: Use Latin characters for file names on attached files.

JAXB cannot parse GX model generated XSD (PL-13598)

Issue: The GX model generated XSD cannot be parsed by JAXB.

Workaround: Add JAXB annotation elements to the XSD to specify the necessary metadata (such as class names) for JAXB to generate the java class files. Contact Guidewire Support for an example XSD annotated in this way.

Cannot make a field from a delegate into a localized column (PL-13761)

Issue: You cannot make a field from a delegate into a localized column.

Workaround: Move the column that you want to localize off the delegate and onto each of the implementing entities. Then, in order to make the column appear as though it exists on the delegate, define on enhancement property on the delegate, that *delegates* to the appropriate column, depending on the implementing entity.

Cannot deploy custom web.xml file (PL-14482)

Issue: BillingCenter does not deploy a custom web.xml file properly.

Workaround: Before attempting to deploy the application, do the following:

1. Open the following file for editing:

BillingCenter\modules\ant\public\gw\ant\deploy.gs

2. Search for the following lines of code:

```
var deployDir = module.Dir.file("deploy")
if (deployDir.exists()) {
    new Copy() { :FileSet = deployDir.fileSet(), :ToDir = destDir }
    .execute()
}
```

3. Make the following change:

```
var deployDir = module.Dir.file("deploy")
if (deployDir.exists()) {
    new Copy() { :FileSet = deployDir.fileSet(), :ToDir = destDir, :Overwrite = true }
    .execute()
}
```

Notice that you need to add :Overwrite = true to the Copy method.

Cannot print second-level list view (PL-14640)

Issue: Printing a second-level list view does not work correctly. The print job contains duplicates and does not include all of the items.

Workaround: Remove the lower level list view out of the hierarchical structure and include it at the top level.

Superuser role unable to edit or delete calendar holidays (PL-14942)

Issue: It is not possible to edit or delete a calendar holiday as superuser.

Workaround: Log into BillingCenter using su to be able to edit the holiday schedule.

RowSet and RowIterator configured in the same ListView do not work correctly. (PL-14946)

Issue: Configuring a RowSet and RowIterator in the same ListView does not provide the right behavior for editability.

Workaround: Separate the RowSet and RowIterator that exist in the same ListView into separate components or into multiple ListView widgets.

Masked input fields do not correctly handle ENTER keypress (PL-14955)

Issue: Input fields that have a field validator mask do not correctly handle the ENTER keypress. After you press ENTER, the field no longer displays the input mask and it is not possible to enter any further input.

Workaround: Cancel the page and re-enter the information.



chapter 32

Guidewire BillingCenter 3.0.4 Release Notes

Release 3.0.4.9

Copyright © 2001-2010 Guidewire Software, Inc. All rights reserved. Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, and the Guidewire logo are registered trademarks, and Guidewire Insurance Suite, Guidewire BillingCenter, Guidewire ContactCenter, and Guidewire Studio are trademarks of Guidewire Software, Inc. All other product names are the trademarks of their respective companies.

This product includes information that is proprietary to Insurance Services Office, Inc (ISO). Where ISO participation is a prerequisite for use of the ISO product, use of the ISO product is limited to those jurisdictions and for those lines of insurance and services for which such customer is licensed by ISO.

This documentation is published as Guidewire Confidential. The contents of this documentation, including product architecture details and APIs, are considered confidential and are fully protected by customer licensing confidentiality agreements and signed Non-Disclosure Agreements (NDAs).

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire BillingCenter.

Version Number

This release of Guidewire BillingCenter is 3.0.4.9.

For Standard Reporting, it supports InetSoft StyleReport Enterprise Edition 10.1, 20100113.

Support

This document includes important information about the current release of Guidewire BillingCenter. For assistance with this software release, contact Guidewire Customer Support. Contact support on the Web at <http://portal.guidewire.com>, by email at support@guidewire.com, or by phone at +1-650-356-4955.

Installing This Release

For installation information, see the *BillingCenter Installation Guide*.

Upgrade Issues

When upgrading from a previous BillingCenter 3.0.x version, the upgrade may fail in the following situations:

- if any line items target the policy period negative writeoff T-account
- if there are any entries in the CollateralContext table with a subtype of CollExcessContext or CollPolPerDrawContext

If you encounter any of these issues, please contact Guidewire Customer Support.

During the upgrade to BillingCenter 3.0.4, it is possible to encounter the following issues:

- Running UpgradeCommission batch process after upgrade throws exception (CC-59048, CC-59387)
- Upgrade of database from BillingCenter 2.1.x to 3.0.4 breaks UpgradeItemsDBCheckBuilder consistency check (CC-59340)
- Upgrade of database from BillingCenter 2.1.x to 3.0.4 requires payment plan initialization (CC-59341)S

[Running UpgradeCommission batch process after upgrade throws exception \(CC-59048, CC-59387\)](#)

It is possible for the upgrade process to fail for any ChargeCommission that is earned “On Payment Rec’d” and in which one of the following is true:

- The sum of the Commissions Payable balance and Commission Payment transactions is positive **and** its charge's Paid Amount is less than or equal to 0.
- The sum of the Commissions Payable balance and Commission Payment transactions is negative **and** its charge's Paid Amount is greater than or equal to 0.

If the Payable Criteria for any commission earner is “On Payment Rec’d”, then perform the following steps before running the database upgrade utility. The upgrade utility executes the same queries outlined in the following steps and it fails if either query returns any results.

1. Run the Commission Payable batch process. This sets the commissions to a consistent state.
2. Run the two queries listed in [Query 1](#) and [Query 2](#).
3. If either query returns any results, reverse the earning transactions that were caused by the writeoff.

The end result of step 3 is this:

- If the sum of the Commissions Payable balance and Commission Payment transactions is positive, then Charge Paid amount must be positive.

- If the sum of the Commissions Payable balance and Commission Payment transactions is negative, then Charge Paid amount must be negative.

WARNING Do not run the Commission Payable batch process again before the upgrade because it undoes the actions in step 3.

Query 1:

```

SELECT (SELECT ta.BalanceDenorm
FROM bc_taccount ta
INNER JOIN bc_taccountpattern tap
ON ta.TaccountPatternID = tap.ID
WHERE ppcc.hiddentaccountcontainerid = ta.taccountcontainerid
AND tap.TAccountName = 'Commissions Payable') AS Payable,
(SELECT SUM(ii.PaidAmount)
FROM bc_invoiceitem ii
WHERE ii.ChargeID = c.ID) AS PaidAmount,
(SELECT SUM(t.Amount)
FROM bc_transaction t
INNER JOIN bc_producercontext pc
ON pc.TransactionID = t.ID
INNER JOIN bctl_transaction tl
ON t.Subtype = tl.ID
WHERE pc.PolicyProducerCodeChargeID = ppcc.ID
AND tl.TYPECODE = 'PolicyCmsnPayable') AS PayableTxns, ppcc.id
FROM bc_policyproducercodecharge ppcc
INNER JOIN bc_policyproducercode ppc
ON ppcc.policyproducercodeid = ppc.id
INNER JOIN bc_commissionssubplan csp
ON ppc.commissionssubplanid = csp.id
INNER JOIN bctl_payablecriteria crit
ON csp.payablecriteria = crit.id
INNER JOIN bc_taccount ta
ON ppcc.hiddentaccountcontainerid = ta.taccountcontainerid
INNER JOIN bc_taccountpattern tap
ON ta.taccountpatternid = tap.id
INNER JOIN bc_charge c
ON ppcc.chargeid = c.id
WHERE tap.taccountname = 'Commissions Payable'
AND ta.balancedenorm + ISNULL((SELECT SUM(t.amount)
FROM bc_transaction t
INNER JOIN bc_producercontext pc
ON pc.transactionid = t.id
INNER JOIN bctl_transaction tl
ON t.SUBTYPE = tl.id
WHERE pc.policyproducercodechargeid = ppcc.id
AND tl.typecode = 'PolicyCmsnPayable'), 0) < 0
AND crit.typecode = 'paymentreceived'
AND ISNULL((SELECT SUM(ii.paidamount)
FROM bc_invoiceitem ii
WHERE ii.chargeid = c.id), 0) >= 0

```

Query 2:

```

SELECT (SELECT ta.BalanceDenorm
FROM bc_taccount ta
INNER JOIN bc_taccountpattern tap
ON ta.TaccountPatternID = tap.ID
WHERE ppcc.hiddentaccountcontainerid = ta.taccountcontainerid
AND tap.TAccountName = 'Commissions Payable') AS Payable,
(SELECT SUM(ii.PaidAmount)
FROM bc_invoiceitem ii
WHERE ii.ChargeID = c.ID) AS PaidAmount,
(SELECT SUM(t.Amount)
FROM bc_transaction t
INNER JOIN bc_producercontext pc
ON pc.TransactionID = t.ID
INNER JOIN bctl_transaction tl
ON t.Subtype = tl.ID
WHERE pc.PolicyProducerCodeChargeID = ppcc.ID
AND tl.TYPECODE = 'PolicyCmsnPayable') AS PayableTxns, ppcc.id
FROM bc_policyproducercodecharge ppcc
INNER JOIN bc_policyproducercode ppc
ON ppcc.policyproducercodeid = ppc.id
INNER JOIN bc_commissionssubplan csp
ON ppc.commissionssubplanid = csp.id
INNER JOIN bctl_payablecriteria crit
ON csp.payablecriteria = crit.id

```

```

    INNER JOIN bc_taccount ta
    ON ppcc.hiddentaccountcontainerid = ta.taccountcontainerid
    INNER JOIN bc_taccountpattern tap
    ON ta.taccountpatternid = tap.id
    INNER JOIN bc_charge c
    ON ppcc.chargeid = c.id
    WHERE tap.taccountname = 'Commissions Payable'
    AND ta.balancedenorm + ISNULL((SELECT SUM(t.amount)
        FROM bc_transaction t
        INNER JOIN bc_producercontext pc
        ON pc.transactionid = t.id
        INNER JOIN bctl_transaction tl
        ON t.SUBTYPE = tl.id
        WHERE pc.policyproducercodechargeid = ppcc.id
        AND tl.typecode = 'PolicyCmsnPayable'), 0) > 0
    AND crit.typecode = 'paymentreceived'
    AND ISNULL((SELECT SUM(ii.paidamount)
        FROM bc_invoiceitem ii
        WHERE ii.chargeid = c.id), 0) <= 0

```

[Upgrade of database from BillingCenter 2.1.x to 3.0.4 breaks UpgradeItemsDBCheckBuilder consistency check \(CC-59340\)](#)

Issue: Upgrading the BillingCenter database from 2.1.x to 3.0.4 breaks the `UpgradeItemsDBCheckBuilder` consistency check. If this happens, you see the following error message:

```
One or more Invoice Items found with type 'Upgrade Commission'.
Please run the UpgradeCommission Work Queue.
```

Workaround: After upgrading the database from BillingCenter 2.1.x to 3.0.4, run the following batch process:

```
UpgradeCommission
```

The process determines how to distribute direct bill and non-primary agency bill commission from the “Upgrade Commission” invoice items to the item level on the actual invoice items. It then removes all “Upgrade Commission” invoice items. You need to run `UpgradeCommission` a single time only after the upgrade triggers have been run.

[Upgrade of database from BillingCenter 2.1.x to 3.0.4 requires payment plan initialization \(CC-59341\)](#)

Issue: Upgrading the BillingCenter database from 2.1.x to 3.0.4 requires that you initialize the payment plan field of policy billing instructions.

Workaround: After upgrading the database from BillingCenter C 2.1.x. to 3.0.4, run the following batch process:

```
UpgradeBillingInstructionPaymentPlan
```

The process initializes the payment plan field of policy billing instructions.

InetSoft Evaluation

You can evaluate InetSoft Style Report Enterprise Edition Version 10.1 using the following license key:

```
L000-72C-ERX-0000BF00100001F-F2B025280327
```

If you want to use InetSoft and the BillingCenter reporting module in a production environment, contact Guidewire Customer Support to request a full license key. For more information, refer to the *BillingCenter Reporting Guide* documentation.

WARNING The build number of the reporting install files must match the build number of the Guidewire BillingCenter installation. If the build numbers do not match, you will have problems with your reporting installation.

Changes in this Release

This section describes the product changes in this release.

- Base PCF File Changes
- Improvements to Upgrade Diff Report
- Improvements and General Issues

Base PCF File Changes

All links below require the `ReleaseNotes_files` directory on your local disk.

BillingCenter release 3.0.3 to 3.0.4

- To view a report of the changes in the base PCF files in the `modules/bc` directory, [click here](#).
- There are no changes to the base PCF files in the `modules/p1` directory.

Improvements to Upgrade Diff Report

With previous releases, Guidewire provided a static report in the release notes detailing certain differences between the current release and the prior minor release. This report described changes in display keys, entities, typelists, and the Gosu API. Guidewire has improved this report to dynamically conform to each customer's particular upgrade path. In addition, the report is regularly updated as the tools for generating it are enhanced to provide more information.

Because the new report is tailored to your particular upgrade requirements, it is no longer included with the general release notes. To obtain your custom Upgrade Diff Report, contact your Guidewire representative. In the future, this report will be available on the Guidewire Resource Center web portal, allowing you to view the latest version right when you need it.

Improvements and General Issues

The following are the improvements added to and the general issues corrected in this release:

ID	Description
Account	
CC-59122	<p>Fixed an issue in which the Funds Transfer Reversal authority limit was not being enforced. The <code>xferfundsreversal</code> authority limit is now used to determine whether a user has sufficient authority to reverse a funds transfer. This single authority limit is used as the limit for reversals of all four types of funds transfers:</p> <ul style="list-style-type: none">• account to account• account to producer• producer to account• producer to producer
Agency Bill	
CC-59221	<p>BillingCenter transactions have wrapper entities that facilitate presentation, authority limit checking, approval process, and so on. At times, wrapper entities can contain multiple transactions. Typically, BillingCenter does not expose the individual transactions to be reversed. You have to reverse the wrapper entity, which reverses all the transactions it contains. However, through configuration, it is possible that transactions are reversed without reversing the wrapper entity.</p> <p>The request for this issue was to reverse the wrapper entity whenever a transaction is reversed. However, there could be multiple transactions on the wrapper entity in which some are reversed and others are not. As an alternative, a new field was introduced on the wrapper entity that stores the reversed amounts. This allows you to write code to compare the original amount and reversed amount to determine whether all the transactions under a wrapper are reversed.</p> <p>In case of <code>AgencyPmtDiffWriteoff</code>, since there are two types of transactions, there are two balances:</p> <ul style="list-style-type: none">• <code>ReversedAmount</code> (for gross writeOff that is reversed)• <code>ReversedCommissionAmount</code> (for reversed Commission WriteOff Amounts, only available for <code>AgencyPmtDiffWriteoff</code>).

ID	Description
CC-59324	Fixed an issue in which a payment reversal would leave an erroneous promise exception for promise items having a disposition of Carry Forward.
CC-59327	If executing a payment results in payment exceptions, and that payment is subsequently reversed, split, or moved, then the exceptions still exist. To modify this behavior (for example, make the payment exceptions disappear), you can configure the Preupdate rules in Studio.
CC-59344	Fixed an issue in which modifying and then reversing an agency bill payment left the original promise unpayable.
CC-59348	Fixed an issue with recalculating totals in the Start Promise wizard and the Start Payment wizard. Previously, if you modified the commission percentage in either wizard and then clicked Recalculate Totals, the % field was reset to the original percentage. Now, BillingCenter shows the correct percentage.
CC-59366	It is no longer possible to have two promises that are active against a single producer statement. For example, if you make a promise, pay it, make a second promise against the same statement, and then reverse the payment, then BillingCenter does not reactivate the original promise because there is already an active promise.
Charge/Invoice	
CC-59302	Previously, the <code>InvoiceItemType</code> typekeys for commission adjustment and commission remainder both had the name "Commission Adjustment". The display name for commission remainder has been changed to "Commission Remainder".
CC-59354	Previously, when default invoice items that were passed into the <code>createFullSetOfInvoiceItems</code> method of the <code>PaymentPlan</code> plugin function were discarded (that is, not returned), during policy cancellation, BillingCenter would mistakenly attempt to examine those items, resulting in an exception. This has been fixed so that discarded invoice items are completely ignored.
Commission	
CC-59104	Fixed an issue with reversing payments after a retroactive producer transfer. Specifically, an <code>IllegalArgumentException</code> occurred if the source producer code had an earning criterion of "On Payment Received" and the destination producer code had an earning criterion of "On Bind".
Disbursement	
CC-59195	Added a new <code>Disbursement</code> plugin and one function (<code>shouldAddAmountToOldestPendingDisbursement</code>) to allow you to configure the behavior of the Automatic Disbursement batch process. The plugin is called if a there is an available amount to disburse and a pending disbursement exists. The default plugin behavior is to add the new amount to the oldest pending disbursement.
CC-59216	The automatic disbursement calculation now works correctly when there are multiple pending disbursements. Previously, in the presence of multiple pending disbursements, the amount of a new disbursement was calculated incorrectly, reflecting only the oldest pending disbursement. This resulted in a negative balance in Account Unapplied.
General Integration	
CC-59226	Guidewire has unretired the <code>integadmin</code> integration administrator permission, which is required to call <code>IMessagingToolsAPI</code> .
Miscellaneous	
CC-58814	You can now reassign the default payer (<code>OverridingDefaultPayer</code>) for a commissionable charge to a producer that is not the primary producer of the policy. Previously, reassignment of the charge payer to a producer was restricted to the primary producer of the policy.
CC-59196	Added a method (<code>shouldMakeInvoiceItemsDueBeforePaying</code>) to the <code>Invoice</code> plugin to allow you to make invoice items due before they are paid. The method is passed in a list of all <code>PaymentItems</code> that are about to be executed. All non-due direct bill invoice items that are returned from this call are moved from their current invoices and placed on a new ad hoc invoice. The new invoice is immediately made billed and due before the execution of the payment completes. This means that the account's Past Due amount and related calculated balances show expected values. Previously, prepaying invoice items (especially negative invoice items) resulted in unexpected denormalized amounts.
CC-59213	Reduced the time it takes to access the Agency Bill Payment wizard from the Agency Bill Payment and Agency Bill Promise pop-ups. This change improves the performance of making a payment on a billed statement invoice, making a payment on a planned statement invoice, and making a promise.
CC-59233	Deprecated the <code>shouldHoldForAgencyBill</code> method on the <code>Invoice</code> plugin.

ID	Description
CC-59259	Fixed an issue in which it was not possible to automatically apply money from a suspense payment to a promise. Now, you can use the <code>Promise.executeAsPayment</code> method to apply money received to a promise even if the money received was created in the same bundle.
CC-59274	Fixed an issue in which the calculation of the amount for <code>Account.AvailableTableForDisbursement</code> ignored the in-memory version of invoice items. Now, the calculation always gets the most current invoice items.
CC-59305	Fixed an issue in which assigning a charge's default payer to a producer caused a null pointer exception if the policy period's producer was null. As a result of this fix, if you assign a producer as the charge's default payer, you must use the <code>ProducerCode</code> for the <code>Charge.OverrideDefaultPayer</code> . Previously, the producer was used instead.
CC-59409	Fixed an issue in which changing the payment plan for an agency bill policy with Refactor All Invoice Items selected and Redistribute Payments set to Yes could generate invalid payments if the policy had existing payments.
CC-59445	Guidewire intends to remove <code>Invoice.PaidStatus</code> in post BillingCenter 3.0.x releases. In preparation for this change, Guidewire deprecates the <code>PaidStatus</code> field on <code>Invoice</code> . Guidewire does not guarantee the data in the <code>Invoice.PaidStatus</code> field for account invoices. In the case of upgraded data, BillingCenter sets the field with different logic and sets the field with the default value during data loading.
Performance	
CC-59219	Fixed an issue in which reversing an agency bill payment against a large statement took too long. Previously, the more payments that were made against a statement, the slower any one payment would take to reverse. Now, no matter how many payments are made against a statement, reversing a payment takes only as long as if there were just one payment.
CC-59227	Improved the efficiency of producer commission calculations. This fix combines three queries into one, thereby reducing round-trips to the database.
CC-59251	The <code>ChargeProRataTx</code> batch process now achieves significantly better throughput. For a given account, <code>ChargeProRataTxns</code> are now processed in a single bundle and in order of <code>EarnDate</code> . To achieve maximum performance benefit, you need to tune the number of distributed worker instances assigned to the <code>ChargeProRataTx</code> batch process.
CC-59280	Improved the performance of <code>Invoice.makeBilled</code> and <code>Invoice.makeDue</code> .
Payment	
CC-59334	When distributing money received to a policy period, BillingCenter now pays all eligible items on the policy period regardless of the payer. Previously, BillingCenter restricted distribution to items on the policy period with their payer set to the account making the payment.
Payment Plan and Invoicing	
CC-59201	Fixed an issue in which an exception occurred during a payment plan change for a policy period with commissable charges if the <code>createFullSetOfInvoiceItems</code> method in the <code>PaymentPlan</code> plugin had been configured to return a completely new set of invoice items. The exception did not occur if the configuration simply modified the default invoice items that were passed to the method.
Platform	
PL-10701	Improved performance with regards to the amount of time that it took to create a new class in Studio.
PL-10812	Corrected an issue in which a finder in a code annotation caused Studio to throw an error, and fail to perform syntax error highlighting and code completion. This occurred even if Studio was connected to the application server (meaning the database).
PL-11508	Corrected a regression that hid a number of methods previously available on <code>Java.external.entity</code> interfaces. Guidewire has re-exposed these methods.
PL-11585	Corrected an issue in which the application did not recognize a modified version of <code>web.xml</code> , but instead utilized the original version.
PL-11931	Updated the description of properties <code>RateScale</code> and <code>NormalizedRate</code> on the <code>ExchangeRate</code> entity to consider them deprecated. Use the <code>Rate</code> property instead to set exchange rates.
PL-12056	Corrected the unit scale on the (Server Tools) <code>Cache Info</code> page. The page displayed the space retained in bytes while the graph label listed the units as kilobytes.
PL-12255	Corrected an issue that could potentially cause a thread deadlock in the <code>GeocodeUtils.getCurrentGeocodePlugin</code> method.
PL-12374	Fixed an issue affecting custom batch processes where the <code>isFinished</code> property was not being set correctly upon process completion.

ID	Description
PL-12464	Fixed an issue that caused an 'Attempt to access bean of type "GroupUser" with a null bundle' error when retrying the assignGroupDynamically and the assignUserDynamically assignment methods after a Validation warning.
PL-12629	Added 0 as a valid value for the attribute samplingpercentage in <databasestatistics> element (used with the <database> element in config.xml). It translates to a statistics estimation percentage of DBMS_STATS.AUTO_SAMPLE_SIZE in Oracle and 100% or FULLSCAN in Microsoft SQL Server.
PL-12877	Fixed an issue that caused a Null Pointer Exception and Studio to stop responding. This occurred if you attempted to create a new rule category with the same name, but different case, of an existing rule category.
PL-12896	Removed an extraneous value from .../webapps/cc/resources/Ocean/css/trans.css.
PL-12989	Fixed an issue that caused the application to not be able to the load root group extension definition during initial upgrade of an empty database or from the import_tools command.
PL-13486	In Studio, as you switch between tabs, close tabs, press ALT-F1 to find in the project tree, and perform other related actions, Studio performs a search to find the node in the project tree that corresponds to the active editor. This fix improves performance in Studio as you switch between tabs, close tabs, and perform similar actions.
PL-13623	Fixed an issue where the PCF editor object was not garbage collected after the PCF editor window was closed. This could cause sluggish Studio behavior or possibly even out-of-memory issues in long Studio sessions in which you created or modified a large number of PCF files.
PL-13626	Improved performance with respect to creating new classes, new PCF files, a new rule set, and new rules in a new rule set in Studio.
PL-13638	Added the following algorithm to implement the XML simple value EncodedString: <pre>if (c < 0x20) { switch (c) { case 0x9: /* Horizontal tab */ case 0xA: /* Line feed */ case 0xD: /* Carriage return */ sb.append(c); continue; default: sb.append('^').append((char)('@' + c)); } } else if (c == '^') { sb.append('^').append(' '); } else if (c == 0xFFFF) { sb.append('^').append('{'); } else if (c== 0xFFFF) { sb.append('^').append('}'); } else { sb.append(c); }</pre> The following URL provides the specification: http://www.w3.org/TR/REC-xml/#charsets
PL-13733	Fixed an issue that cause a CDCE exception (ConcurrentDataChangeException) due to stale cache data.
PL-13793	Fixed an issue that caused a Version List to contain incorrect effective dates if the effective date of a listed policy was later than the expiration date of the policy in the row above it.
PL-13809	Added the ability to access generated XSD files used for archiving at the following location: http://servername:port/xx/service/xsd For example, for BillingCenter, this is: http://localhost:8580/bc/service/xsd
PL-14009	Fixed an issue in which GX models containing subtype mappings did not show all the subnodes descendant to the subtype node.
PL-14091	Fixed an issue in which a Claim search using the "Any Party Involved" option did not return the correct response if the value entered for Organization Name could match a Place name.

ID	Description
PL-14130	Corrected an issue that caused policies with a large number of data entities (such common commercial policies) to experience significant performance issues. Policies with root entities that had a large number of child entities (a location that contained a large number of buildings, for example) were especially vulnerable to this type of performance issue.
PL-14214	Reinstated the requirement that a Studio user must have the Administer Rules permission (with code ruleadmin) to access a running production server.
PL-14389	Corrected an issue with SQL Server that occurred during recreation of primary keys with backing clustered indexes during an upgrade. The process did not account for self-referencing foreign keys as it dropped foreign keys prior to recreating the primary keys.
PL-14418	Corrected an issue that caused a report to freeze if that report contained free-hand tables along with large data sets. This issue was due to a bug in the InetSoft v9 reporting software.
PL-14446	Corrected an issue that occurred if moving to clustered indexes on SQL Server during an upgrade and using filegroups, the application moved the data from the original filegroup. The application now leaves the data in the original filegroup when moving to clustered indexes during an upgrade.
PL-14483	On application startup, BillingCenter checks the database version against the required major version and minimum minor version. BillingCenter generates an error if the major version does not match or if the minor version is less than the minimum minor version supported. This is a change from previous (pre-7.0.0) releases as later minor release versions would also generate an error. See the "Database Patch Strategy 2011" white paper or the Platform Support Matrix available on the Guidewire Resource Portal for more information.
Producer	
CC-59060	Fixed an issue that blocked the reversal of a funds transfer between two producers.
CC-59094	Added an option to the OutgoingProducerPaymentReversalConfirmationPopup to control whether to reverse PaymentSent and all associated commission payable transactions or reverse only the PaymentSent transaction. Reversing only the PaymentSent transaction keeps commission money at the producer level. This allows the reversed funds to be transferred to another producer (or to follow normal commission payable workflow and be sent out in a future payment). The default for the Reversal Option is to reverse both the PaymentSent transaction and associated commission payable transactions. The result is that the commission money is pushed back to the charge, policy, or item level. The reversal option was introduced to fix an issue in which an exception occurred if you reversed a producer payment through the user interface and then attempted to transfer the commissions payable to another producer.
CC-59096	Fixed an issue in which the Producer Statement screen that displays direct bill commission for a producer displayed incorrect information regarding any CommissionPayable transfers that occurred on the producer. BillingCenter now correctly displays information about CommissionPayable transfers from one producer to another.
Reinstatement Process	
CC-52391	Fixed an issue in which the reinstatement fee was not being charged as BillingCenter executed a Reinstate- ment billing instruction. The fix required changing the signature on the getReinstatementFeeAmount method in the Invoice plugin to take the delinquency plan and delinquency target.
Suspense Payment	
CC-59076	You can now reverse a suspense payment that has been disbursed. For example, a customer sends in a check, but the relevant account or policy has not yet been created, so a suspense payment is created. The customer then changes his or her mind and cancels the policy before it is issued, resulting in a disbursement. However, the customer check that originally engendered the suspense payment bounces (after the suspense payment has been disbursed), resulting in the need to reverse the suspense payment.
CC-59238	Added a new method on SuspensePayment with the signature applyTo(Account payer, PolicyPeriod policyPeriod), which applies the suspense payment to the policy period. The payer of the DirectBillMoneyRcvd returned by the method is the account that is passed to the method. The DirectBillMoneyRcvd is distributed to invoice items that belong to the policy period. If you want the account on the DirectBillMoneyRcvd to affect the distribution, you can achieve this by overriding the IDirectBillPayment plugin.
Staging Table	
CC-52820	Credit and CreditReversals are now loadable using the staging tables.

ID	Description
CC-59211	Added a consistency check that ensures delinquency plans have at least one DelinquencyPlanReason with a DelinquencyReason of type PastDue.
CC-59289	Added a consistency check to ensure that only one non-reversal ChargePaidFromAccount or ChargePaidFromProducer transaction exists for a BasePaymentItem.

Known Issues and Limitations

Note: For maintenance releases, it is Guidewire policy to defer fixing configuration issues if they require merging files during the upgrade. Workarounds to many of these issues are listed in the following sections. The goal of this policy is to make upgrades as straightforward as possible.

This section describes known issues with this release of Guidewire BillingCenter.

BillingCenter displays many warnings when regenerating Javadoc (PL-10829)

Issue: This issue occurs when running the `regen-java-api` and `regen-soap-api` command.

Workaround: Guidewire is aware of this issue.

Incorrect additivity statements in the default `logging.properties` file (CC-50615)

Issue: Some additivity statements in the `logging.properties` file are incorrect and therefore, ineffective. An additivity statement attempts to keep the category from sending its log messages up to its parent, preventing duplicate log messages across files.

Workaround: In the `Logging.properties` file (located at `modules/cc/config/logging`), change:

```
log4j.additivity.PluginsLog=false
to
log4j.additivity.Plugin=false
```

Problems occurred with the use of negative collateral charges (CC-51311)

Issue: When issuing a negative collateral charge, the amount gets subtracted from a Charge Due T-account and a payer's Unapplied T-account. This causes an imbalance between `collateral_reserve` and `collateral_held` when issuing a negative charge.

Workaround: Guidewire is aware of this issue.

Automatic distribution to policies not working correctly (CC-51511)

Issue: Automatic distribution to policies on `Account → Disbursement` screen does not distribute the payment correctly.

Workaround: Guidewire is aware of this issue.

Billing instructions for Issuance and Policy Change must work with premium and other charges (Premium Reporting Only) (CC-52032)

Issue: This issue is for Premium Reporting policy periods only. Users are note allowed to enter premium charges as part of an issuance or policy change and process them as any other policy period. This issue applies to all billing instructions types.

Workaround: Guidewire is aware of this issue.

No commission reserve and onset transfer transaction was created when a direct bill item was reassigned to an agency bill producer (CC-54923)

Issue: When a direct bill policy has no specified primary producer and one of the policy's item is reassigned to an agency bill producer, no commission reserve and onset transfer transaction created. This was because there is no initial reserve.

Workaround: Guidewire is aware of this issue.

New collateral charges and invoice amounts are being generated based on previously paid collateral amounts, not the previously billed or planned amounts (CC-55209)

Issue: New collateral charges and invoice amounts are being generated based on the amount of collateral previously paid and do not account for amounts of collateral previously Billed or Planned (not yet billed). This results in an over billing of the collateral amount on the invoices.

Workaround: Guidewire is aware of this issue.

Errors occur in the totals for the producer's rate on the Transactions screen (CC-55835)

Issue: A difference occurs in the totals shown on the **Transactions** screen for a producer when selecting **Reserves** and then choosing either the **Last 30 Days** or **Show All** filter.

Workaround: Guidewire is aware of this issue.

Account Evaluation is not counting account level delinquencies (CC-55938)

Issue: The count for number of delinquencies on **AccountEvaluation** screen doesn't factor account level delinquencies.

Workaround: Use the Gosu finder to identify the account level delinquency processes and add it to the BillingCenter value.

After a reversal, the distributed payment is still being reserved for the premium payment (CC-56538)

Issue: The disbursement amount is not fully moved to disbursement and does not show up on the **Disbursement** page although an entire payment has been reversed and the associated charge had been assigned to another account.

Workaround: This is a new feature that is not fully implemented. Guidewire is aware of this issue.

Moving a Billed invoice item should negate the direct bill commission paid, not the agency bill commission paid (CC-56656)

Issue: If an item is moved from a **Billed** invoice to a **Planned** invoice, an earnings transaction is created for the Payable amount on the commission. However, if the commission was earned and paid, it is not being negated.

Workaround: Guidewire is aware of this issue.

BillingCenter did not process policy changes done after the policy expiration date (CC-56852)

Issue: During BillingCenter - Policy Center integration, BillingCenter does not process policy changes made after the policy expiration date. An illegal state exception shows on both the BillingCenter and PolicyCenter consoles.

Workaround: Guidewire is aware of this issue.

[Incorrect totals in the policy payment schedule occurred on agency bill statements after a change was made to one of two producers on a policy \(CC-56925\)](#)

Issue: The producer statement does not show the correct dollar amount after a change is made to one of the policy's payment plans on a policy with two producers after the statement was billed and partially paid.

Workaround: Guidewire is aware of this issue.

[Duplicate payment plan database constraint on starting an integrated environment and sending cancelPolicy message \(CC-56972\)](#)

Issue: BillingCenter does not process policy changes done after the policy expiration date. Illegal state exception is received in console of both BillingCenter and PolicyCenter.

Workaround: Guidewire is aware of this issue.

[Should not be able to edit Payment Plan after it has been used by a policy \(CC-57758\)](#)

Issue: In the base configuration, it is possible to edit a Payment Plan even after it has been used by a policy. This is incorrect. Only the expiration date should be updateable.

Workaround: Configure the payment plan screen to only allow the expiration date to be updated.

[Invoice PreUpdate rule is bypassed when a new invoice created for Collateral charge \(CC-58254\)](#)

Issue: Automatic distribution to policies on Account → Disbursement screen does not distribute the payment correctly.

Workaround: Guidewire is aware of this issue.

[No API to remove invoice items from DirectBillPaymentView \(CC-58397\)](#)

Issue: There is no way to programmatically remove an Invoice item (or invoice item array) from a distribution.

Workaround: Guidewire is aware of this issue.

[Use of StringBuilder Class prevents debugging of web service \(CC-58533\)](#)

Issue: In a web service, if you have a method with a parameter of type X and inside the web service method you reference a Gosu class of type Y. And, Gosu class Y has a `toString()` method (overriding the default) that uses a `java.lang.StringBuilder` class. If you set a breakpoint in the web service method and try to step through the method, then you will get errors.

Workaround: Guidewire is aware of this issue.

[Reversing Write-Off Commissions and Write-Off Commission Recovery in the UI \(CC-58655\)](#)

Issue: The BillingCenter user interface does not provide a way reverse Write-Off Commissions and Write-Off Commission Recovery.

Workaround: Clone the Reverse button under the Payments tab and reverse `CmsnPayableReduction` and other transactions using the following action: `CurrentLocation.startEditing(); producerTransaction.reverse();` and `CurrentLocation.commit();`.

[Cannot un-distribute direct bill payment \(CC-58764, CC-58764\)](#)

Issue: The base configuration does not allow a direct bill payment to be undistributed.

Workaround: Use the configuration to create an **Unapply** button on the **Payment List View**. This button should remove the original **DirectBillMoneyRcvd**, reverse all associated transactions, and then create a brand new **DBMoneyRcvd** with the same attributes as the original.

Amount distributed on a payment does not reflect transfers.(CC-58782)

Issue: The Amount Distributed on a payment does not reflect transfers. This means that the **Account : Unapplied** is adjusted, but the full payment amount is still available for distribution on the original account.

Workaround: Make the following amendments to the **allocatePayment** method of the **DirectBillPayment.gs** plugin. Add the following lines to adjust the amount to allocate before you call out to the allocation strategy:

```
allocatePayment( DirectBillPayment, Amount)
if ( payment.BaseMoneyReceived.Executed) {
    amount = BigDecimalUtil.min(payment.Account.RawUnappliedAmount, amount)
} else {
    amount = BigDecimalUtil.min(
        payment.Account.RawUnappliedAmount.add(payment.BaseMoneyReceived.Amount),
        amount)
}
```

The same issue happens if you call the following plugin methods:

- **allocateCollateral**
- **allocatePolicyPeriod**
- **allocateCredits**.

You can add the same lines of code to these plugin methods to correct the problem that occurs if the payment you are allocating has more money in it than what is actually executed against **Account : Unapplied**.

On AIX without an X11 server, BillingCenter may report that an X11 resource is missing (CC-5889)

Issue: BillingCenter does not require an X11 server; however other components of the BillingCenter environment (such as Java) may inadvertently have this requirement. If so, BillingCenter may report that an X11 resource is missing when it is starting up.

Workaround: Install an X11 server, or contact Guidewire support to determine if the dependency can be removed.

Changing commission rate after assigning charge's payer to Producer fails with *EvaluationException* (CC-59301)

Issue: If you reassign the payer of a charge from an account to a producer and subsequently override the commission rate, then the override fails with *EvaluationException*.

Workaround: Perform the commission rate override first. Then, reassign the charge payer to the producer. This sequence calculates the commission rate correctly for the assigned items and BillingCenter does not throw an exception.

Consistency checks on databases with a large numbers of accounts can take a long time to complete (CC-59330)

Issue: Consistency checks on databases with over 100,000 accounts can take longer than expected. In internal tests, consistency checks on a database with 100,000 accounts completed in 4 minutes. On a database with 1,000,000 accounts, consistency checks took 25 hours.

Workaround: Guidewire is aware of this issue.

Report folders do not display if empty (PL-1281)

Issue: Guidewire BillingCenter does not display a report folder if that folder is empty.

Workaround: Place an InetSoft replet inside the folder. You can mark the replet as *not visible* within the InetSoft Enterprise Manager. This ensures that the parent folder displays in BillingCenter even if the folder is empty.

Acrobat Sample document template does not allow extra fields (PL-1416)

Issue: Currently, the example Acrobat document template uses two fields, and this requires the same number of fields in `SampleAcrobat.pdf.descriptor` and `SampleAcrobat.pdf` files.

Workaround: To allow more fields in the template and documents created from it, you must update its descriptor file.

Studio PCF editor treats commented-out property values as errors (PL-4582)

Issue: If you attempt to comment out a property value in the Studio PCF editor, Studio treats this as an error.

Workaround: Guidewire is aware of this issue.

BillingCenter not generating `soap.local.entity` correctly (PL-7560)

Issue: BillingCenter is not generating the `soap.local.entity` for a class used by web services.

Workaround: To ensure that BillingCenter generates the `soap.local.entity` correctly for `FieldValue` classes (which are Java types), subtype `FieldValue` into `GosuFieldValue`, then further subtype that class.

Server exception during startup (PL-8167)

Issue: During server startup, there can be a Tomcat server exception. However, after the exception, the server does start up. This is caused by the Tomcat application server attempting to either save or restore sessions using serialization. The session restore causes an exception because BillingCenter sessions contain objects which rely on the metadata being started up.

Workaround: None. The Tomcat application server does start up after the error.

Zone data with missing values cannot be loaded (PL-9123)

Issue: Zone data, which is used for associating postal codes with cities and regions, does not load if it is incomplete.

Workaround: Guidewire is aware of this issue.

H2 development database creates LONGTEXT fields as VARCHAR(65000), instead of as CLOB as in Oracle and SQL Server (PL-9314)

Issue: Guidewire creates schema fields defined as LONGTEXT as LONGVARCHAR columns, instead as CLOB as in the Oracle or SQL Server databases. In addition to the inconsistency involved, this mandates a field length of 65,000 characters or less in LONGTEXT columns in the H2 database.

Workaround: Define a LONGTEXT field as a `VARCHAR(...)` with the required size, for example, as `VARCHAR(120000)`.

Tab widgets do not work with Microsoft Windows Server 2003 enhanced security enabled (PL-9813)

Issue: If you have the Windows Server 2003 component called IE Enhanced Security Feature installed, tabs may be unresponsive in the user interface.

Workaround: Uninstall this component from the Windows 2003 Server.

Not possible to hot-deploy on all supported application servers (PL-9955)

Issue: It is not possible to hot-deploy on all supported application servers.

Workaround: If you use Tomcat, you are able to see the changes without requiring a restart of the application server. If you use an other application server, you need to perform a restart to see any configuration changes reflected.

[First time you click on the arrow of the typekey input, the drop-down menu will not open \(PL-10134\)](#)

Issue: The drop-down menu does not open on the first click of the arrow on a typekey input. Instead, the help text opens.

Workaround: Turn off help text on focus by setting `InputHelpTextOnFocus` to `false` in the `config.xml` file. By doing that, the help text shows only if you mouse over the input and will not interfere with opening a drop down menu.

[InetSoft exception Failed Login Exception: Bad user name anonymous \(PL-10148\)](#)

Issue: InetSoft generates a `Failed Login Exception: Bad user name anonymous` exception on server start-up. This occurs if you set the InetSoft logging level to `fine`.

Workaround: This error does not cause any issues. You can safely ignore the exception.

[Report still visible after permissions disabled for parent folder \(PL-10353\)](#)

Issue: An InetSoft report that is included in a permission set is still visible in the report tree even after its parent has been removed from the permission set.

Workaround: You must remove individual reports from permission sets instead of maintaining the reports at the report folder level.

[Inability to access typecodes that contain a dash \(PL-10532\)](#)

Issue: It is not possible to access a typekey code (typecode) that contains a dash using Gosu.

Workaround: Remove any dashes from the typekey codes.

[Studio Debugger does not handle rule breakpoints properly \(PL-10553\)](#)

Issue: If you set a breakpoint in a Gosu rule and attempt to step through the code using the Studio debugger, the debugger does not correctly display the code on the breakpoint step.

Workaround: Guidewire is aware of this issue.

[ToolbarButton widget does not render correctly \(PL-10742\)](#)

Issue: BillingCenter incorrectly renders a `ToolbarButton` widget if the associated `AddMenuItemIterator` is empty.

Workaround: Modify your configuration to disable the menu button if the button contains no menu item.

[Must name row iterator element with capital letter \(PL-10795\)](#)

Issue: The Document search results page (`DocumentLV.pcf`) throws a `NullPointerException` if the row iterator element name is `document` (with a lower case d).

Workaround: You must use `Document` (with a capital D) instead.

[Studio does not properly report verification Error \(PL-10822\)](#)

Issue: Studio does not properly report a verification error if a typecode contains a trailing space.

Workaround: Remove trailing spaces from all type codes.

Navigating between pages in wizard loses data (PL-10920)

Issue: In a wizard pop-ups, it is possible to lose data if you navigate to another page in the wizard.

Workaround: Navigate out of the wizard and then back into it rather than changing steps directly in the wizard.

Values in MIME-Type drop-down not localized (PL-10921)

Issue: BillingCenter does not localize the values in the **MIME type** drop-down list properly.

Workaround: If you want to localize the MIME type names, you can do so by modifying the **description** attribute on the `<mimetype>` element definitions in `config.xml`. Currently, BillingCenter does not support MIME type definitions in multiple languages.

Studio—with Subversion (SVN)—incorrectly copies .svn files (PL-10932)

Issue: *This issue effects installations that use Subversion only.* If you modify the base configuration rules in Studio, Studio creates a copy of the rule in the `modules/configuration` folder. Studio then executes Subversion commands to add the files and folders to Subversion. In the process, Studio incorrectly copies the `.svn` folder and its children. This leads to Subversion problems.

Workaround: Manually delete the copied `.svn` folders

Double quotes in *Path* environment variable causes problems (PL-10981)

Issue: BillingCenter does not run if there are double quotes in the `PATH` environment variable.

Workaround: Check the `PATH` variable and remove any double quotes.

Large number of classes in `plugins\shared\lib` causes studio to fail to start (PL-11026)

Issue: If you attempt to load multiple JAR files that contain many thousands of classes, Studio fails to start and throws an `OutOfMemory` error.

Workaround: Increase `MaxPermSize` to something over `128m` (the default) so that Studio has enough memory to build the type information database.

ANT command `build-war` hangs indefinitely (PL-11045)

Issue: It is possible for the `build-war` ANT command to hang indefinitely. This issue is only reproducible on certain environments.

Workaround: Set the `-noinput` parameter before calling the `build-war` ANT command. For example, instead of `gwcc.bat build-war`, do the following:

```
gwbc.bat -noinput build-war
```

Cache summary statistics rely on server ID being unique (PL-11239)

Issue: If you use the same server ID for multiple servers in a cluster, it can create the following problems:

- Different servers in a cluster deleting or changing each others results
- Deadlocks at the database level

Workaround: By default, the server ID defaults to the host name of the machine running the instance. This works well in many configurations, but causes problems if multiple server instances are running on a single machine. Set the `serverid` system property in the `config.xml` file on the local server host to a unique value in the cluster. If it does not currently exist, then you need to add a `<registry>` element to the `config.xml` file on the local server host.

You can also start the server with the following JVM option: `gw.bc.serverid=????`

See *Defining the Application Server Environment* in the System Administration Guide for details on both of these options.

[Missing display keys cause server to fail to start \(PL-11764\)](#)

Issue: Missing display keys defined in the application default locale can cause the server to fail to start.

Workaround: Within Studio, run the **Verify Resources** utility before attempting to start the application server. BillingCenter bases the type system for display keys on those defined in the default application locale. If a display key in the default application locale is missing and a rule references the missing display key, then BillingCenter cannot compile the rule. Running the resource verification utility can catch these types of missing display keys.

[Countries configured in zone-config.xml still generate a warning during regen-dictionary even if zone data loaded for all of these countries \(PL-11947\)](#)

Issue: Countries configured in zone-config.xml still generate a warning during regen-dictionary even when zone data is loaded for all of these countries.

Workaround: Warning message is created in error and can safely be ignored.

[The user interface cannot handle starting multiple instances of a batch process \(PL-12372\)](#)

Issue: The user interface cannot handle multiple instances of a batch process.

Workaround: If multiple instances of a batch process need to be executed, they must be started from the command line. Also, you should ensure that the `BatchProcess.isExclusive()` method returns `false` to allow multiple instances to be run simultaneously.

[Debugger break point for the rule engine does not work if first the line is a comment or a blank line \(PL-12912\)](#)

Issue: The debugger break point for the rule engine does not work if the first line is a comment or a blank line.

Workaround: Ensure that the first line is not blank and does not contain a comment. Add comments after the first line.

[Type system refresh after PCF page title change does not update corresponding menu label \(PL-13057\)](#)

Issue: The type system refresh after the PCF page title change does not update corresponding menu label.

Workaround: After updating a page title, the server restart must be done to refresh menu labels and avoid null pointer exceptions due to stale reference.

[Length limitation on entity localization table names \(PL-13360\)](#)

Issue: There is a length limitation on entity localization table names.

Workaround: Ensure that localization `tableName` property specified in the entity extension file is less than 16 characters. The error message generated if the localization table name exceeds the maximum length indicates that 18 characters are allowed, but that does not account for two additional characters added by the application.

Use of `StringBuilder` class prevents debugging of web service (PL-13435)

Issue: Suppose that you have a web service in which you have a method with a parameter of type X and inside the web service method you reference a Gosu class of type Y. This can be an issue if the Gosu class Y has a `toString` method (overriding the default) that uses a `java.lang.StringBuilder` class. If you attempt to set a breakpoint in the web service method and try to step through the method, Studio generates errors.

Workaround: Studio does not generate errors if you call the web service in normal run mode rather than in debug mode.

`JavaToolkit.gs` has incorrectly hard-coded memory that results in failed `regen-java-api` Ant task (PL-13663)

Issue: File `JavaToolkit.gs` has hard-coded memory that can result in failed `regen-java-api` Ant tasks.

Workaround: Increase the size of the max heap setting on line 161 of `JavaToolkit.gs` in the Ant module. The shipped value is 512.

`US-Locations.txt` file with the US geodata from GreatData has special characters that cause validation problems with United States Postal Service (USPS) data (PL-13384)

Issue: The `US-Locations.txt` file contains information that does not conform to United States Postal Service (USPS) standards for bulk mailings.

Workaround: The provided `US-Locations.txt` file is intended only for use in geocoding to identify addresses for a location. You can process the `US-Locations.txt` file to conform to your particular address standards, and then import that version of the file instead.

GX models that reference virtual fields and enhancements throw null pointers if null (PL-13560)

Issue: The GX models that reference virtual fields and enhancements throw null pointers when they are null.

Workaround: Ensure that null checks and error handling is included so that if referenced virtual fields or enhancements are null, then there will not be a null pointer exception.

Sending email with file attachment with unicode filename is not correctly handed over to the mail server (PL-13582)

Issue: Sending email with file attachment with unicode filename is not sent to the mail server correctly.

Workaround: Use Latin characters for file names on attached files.

JAXB cannot parse GX model generated XSD (PL-13598)

Issue: The GX model generated XSD cannot be parsed by JAXB.

Workaround: Add JAXB annotation elements to the XSD to specify the necessary metadata (such as class names) for JAXB to generate the java class files. Contact Guidewire Support for an example XSD annotated in this way.

Cannot make a field from a delegate into a localized column (PL-13761)

Issue: You cannot make a field from a delegate into a localized column.

Workaround: Move the column that you want to localize off the delegate and onto each of the implementing entities. Then, in order to make the column appear as though it exists on the delegate, define an enhancement property on the delegate, that delegates to the appropriate column, depending on the implementing entity.

Copy and paste does not work from phone fields (PL-13792)

Issue: The copy and paste does not work for phone fields on a contact.

Workaround: Save the contact with the newly entered phone information before attempting to either copy or paste the value into another field.

Cannot deploy custom *web.xml* file (PL-14482)

Issue: BillingCenter does not deploy a custom *web.xml* file properly.

Workaround: Before attempting to deploy the application, do the following:

1. Open the following file for editing:

```
BillingCenter\modules\ant\public\gw\ant\deploy.gs
```

2. Search for the following lines of code:

```
var deployDir = module.Dir.file("deploy")
if (deployDir.exists()) {
    new Copy() { :FileSet = deployDir.fileSet(), :ToDir = destDir }
        .execute()
```

3. Make the following change:

```
var deployDir = module.Dir.file("deploy")
if (deployDir.exists()) {
    new Copy() { :FileSet = deployDir.fileSet(), :ToDir = destDir, :Overwrite = true }
        .execute()
```

Notice that you need to add `:Overwrite = true` to the `Copy` method.

Cannot print second-level list view (PL-14640)

Issue: Printing a second-level list view does not work correctly. The print job contains duplicates and does not include all of the items.

Workaround: Remove the lower level list view out of the hierarchical structure and include it at the top level.

ContactRoleTypeConstraint configuration fails if using *ExceptionConstraint* (PL-14748)

Issue: If you use an `ExceptionConstraint` in `entityroleconstraints-config.xml`, BillingCenter does not properly enforce the constraints.

Workaround: Create a separate role for each of the subtypes associated with the given role.

Upgrade from ClaimCenter 4.x to ClaimCenter 6.x upgrades Regions but not associated Region Zones (PL-14761)

Issue: The upgrade from ClaimCenter 4.x to ClaimCenter 6. upgrades Regions. However the upgrade process does not bring the associated Region Zones along with the Regions.

Workaround: Manually add the zones. You can also do a file import of the zones using `import_tools`.

Superuser role unable to edit or delete calendar holidays (PL-14942)

Issue: It is not possible to edit or delete a calendar holiday as `superuser`.

Workaround: Log into BillingCenter using `su` to be able to edit the holiday schedule.

RowSet and *RowIterator* configured in the same *ListView* do not work correctly. (PL-14946)

Issue: Configuring a `RowSet` and `RowIterator` in the same `ListView` does not provide the right behavior for editability.

Workaround: Separate the RowSet and RowIterator that exist in the same ListView into separate components or into multiple ListView widgets.

Masked input fields do not correctly handle ENTER keypress (PL-14955)

Issue: Input fields that have a field validator mask do not correctly handle the ENTER keypress. After you press ENTER, the field no longer displays the input mask and it is not possible to enter any further input.

Workaround: Cancel the page and re-enter the information.

BillingCenter does not recognize modified *Calendar.js* file (PL-14997)

Issue: BillingCenter does not correctly recognize changes made to *Calendar.js* through Studio. You access this file in the following Studio location:

Resources → Web Resources → resources → javascript → global

Workaround: Instead, modify *global.js* directly. You access this file in the following installation file location:

BillingCenter/webapps/bc/resources/javascript

WARNING This is one of the very few occasions in which you modify an application file outside of the application configuration folder. Ensure that you modify *global.js* only. Otherwise, you can invalidate your installation.



chapter 33

Guidewire BillingCenter 3.0.3 Release Notes

Release 3.0.3.14

Copyright © 2001-2010 Guidewire Software, Inc. All rights reserved. Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, and the Guidewire logo are registered trademarks, and Guidewire Insurance Suite, Guidewire BillingCenter, Guidewire ContactCenter, and Guidewire Studio are trademarks of Guidewire Software, Inc. All other product names are the trademarks of their respective companies.

This product includes information that is proprietary to Insurance Services Office, Inc (ISO). Where ISO participation is a prerequisite for use of the ISO product, use of the ISO product is limited to those jurisdictions and for those lines of insurance and services for which such customer is licensed by ISO.

This documentation is published as Guidewire Confidential. The contents of this documentation, including product architecture details and APIs, are considered confidential and are fully protected by customer licensing confidentiality agreements and signed Non-Disclosure Agreements (NDAs).

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire BillingCenter.

Version Number

- This release of Guidewire BillingCenter is 3.0.3.14.
- For Standard Reporting, it supports InetSoft StyleReport Enterprise Edition 10.1, 20100113.

Support

This document includes important information about the current release of Guidewire BillingCenter. For assistance with this software release, contact Guidewire Customer Support. Contact support on the Web at <http://portal.guidewire.com>, by email at support@guidewire.com, or by phone at +1-650-356-4955.

Installing This Release

For installation information, see the *BillingCenter Installation Guide*.

Upgrade Issues

When upgrading from a previous BillingCenter 3.0.x version, the upgrade may fail in the following situations:

- if any line items target the policy period negative writeoff T-account
- if there are any entries in the CollateralContext table with a subtype of CollExcessContext or CollPolPerDrawContext

If you encounter any of these issues, please contact Guidewire Customer Support.

InetSoft Evaluation

You can evaluate InetSoft Style Report Enterprise Edition Version 10.1 using the following license key:

L000-798-ERX-0000BF00100001F-F2AFC9DD34ED

If you want to use Inetsoft and the BillingCenter reporting module in a production environment, contact Guidewire Customer Support to request a full license key. For more information, refer to the *BillingCenter Reporting Guide* documentation.

WARNING The build number of the reporting install files must match the build number of the Guidewire BillingCenter installation. If the build numbers do not match, you will have problems with your reporting installation.

Changes in this Release

This section describes the product changes in this release.

- Base PCF File Changes

Base PCF File Changes

All links below require the `readme_files` directory on your local disk.

BillingCenter release 3.0.2 to 3.0.3

- To view a report of the changes in the base PCF files in the `modules/bc` directory, [click here](#).
- To view a report of the changes in the base PCF files in the `modules/platform` directory, [click here](#).

General Issues and Improvements

The following are the improvements added to and the general issues corrected in this release:

ID	Description
Accounts	
CC-58887	Fixed an issue where an exception occurred when you tried to close an account if the Account Summary screen was in Edit mode. The Close Account button is now disabled in Edit mode.
Activities	
CC-58970	Fixed an issue where a user with the actapproveany permission (to approve any approval activity) was unable to approve an activity that was not owned by the user (even with sufficient authority limits).
Agency Bill	
Payment Plan and Invoicing	
CC-58924	After a payment plan change, you can now specify that any agency bill payments that had existed against invoice items removed by the payment plan change will be redistributed.
Agency Bill	
CC-56153	Fixed an issue where incorrect amounts were displayed in the Amount Due (Net) and Payment Applied (Net) fields on an agency bill statement after a partially paid agency bill item was reassigned to another producer.
CC-58899	Fixed an issue where past due items would be discarded from a statement after an agency bill plan change. These invoice items are now added to the first planned invoice.
CC-58900	Corrected an issue where the overriding commission percentage was incorrectly displayed when you attempted to edit an agency bill promise or payment.
Charge/Charge Group	
CC-58939	Fixed an issue with delinquency not exiting as expected when a policy endorsement negative charge should result in paying down the amount due to within the delinquency threshold. This issue occurred when the InvoiceAssembler plugin was customized to place policy endorsement negative charges on an already billed and due invoice.
Collateral	
CC-58937	You can now allocate a portion of a payment towards a collateral requirement without having to create a charge for it. To support this feature, the Collateral Item card was added to the New Payment screen for direct bill payments. There is a limit of one collateral item for each payment.
Commission	
CC-57106	Fixed issues regarding the retroactive transfer of a producer on a direct bill policy. If the transfer is from a producer code to a null producer, the commission reserve is now removed. If the transfer is from a null producer to a producer code, commission reserve is now created according to the rate for the producer role on the applicable subplan of the target producer code.
CC-58832	Fixed an issue in the computation of commission remainder when the rounding method configuration parameter is "Round Half Even". Another issue was identified and clarified as being as designed. Specifically, different rounding methods are used to establish the initial commission reserve (on a charge) and to establish the commission amount on item commissions. The former uses the configuration parameter RoundingMode specified in config.xml . The latter always uses "Round Down". This means that if the parameter CommissionRemainderTreatment is set to writeoff , the carrier may pay a few pennies less (with a write-off) instead of few pennies more.
CC-59048	It is now possible to specify payable criteria at the charge level. A scriptable field named PayableCriteria has been added to ChargeCommission . The field is nullable. If the PayableCriteria field on the ChargeCommission is set, commission earning for that ChargeCommission takes place based on the PayableCriteria for the ChargeCommission (instead of the PayableCriteria of the CommissionSubPlan , which normally dictates how commission gets earned).
Ad Hoc Credit	

ID	Description
CC-53423	Fixed an issue with the Account Journal screen where incorrect descriptions were displayed for Account Interest and Account Other transactions.
Core-Platform	
PL-11189	Guidewire has determined that an intermittent verification error on autoCompleteArgIds occurs due to a JVM bug. You can safely ignore this error. Or, if you choose, you can upgrade your JVM to 1.6.0_21 or later, which removes the cause of the error message.
PL-11310	Corrected an issue in which, if you were defining the sortOrder for an element related to a query-backed ListView, Studio did not report an error if you chose an element for sorting that could not be used for sorting the backing query.
PL-11572	Corrected an issue that caused a NullPointerException if you canceled out of a page that contained unsaved work and the page contained data that had been deleted in the database.
PL-11691	Added a tool that generates a report listing the Gosu rules defined in the Rules editor in Guidewire Studio. To generate the report, type gwcc regen-rulerreport in the application bin directory. See the System Administration Guide for more information.
PL-12314	Modified the data model and added a new element remove-index to the data model elements. You can use this element to remove an index if it exists purely for performance. Such indexes are either a) non-unique, or b) unique but contain an ID column. (Guidewire performs metadata validation to ensure that the element removes only those indexes that fall into one of these categories.) To modify a database index, you can remove it using this element, then add a new index with a different name and with your characteristics.
PL-12402	Modified the VenueType typelist to make it non-final.
PL-12500	Corrected an issue in which the Data Dictionary did not display certain virtual properties on an entity if the virtual properties were virtual property fields on a delegate implemented by the entity.
PL-13076	Added a tool that generates a report listing the Gosu rules defined in the Rules editor in Guidewire Studio. To generate the report, type gwcc rulerreport in the application bin directory. See the System Administration Guide for more information.
PL-13430	Corrected an issue associated with localizing a column on a custom entity that created a gw.xml.XMLParseException: Could not parse XML file temp.dti exception. The exception occurred if you added a <localization> element to <column> and used an underscore in the column or entity name.
Disbursement	
CC-59015	When the Automatic Disbursement batch process runs and the amount of the automatic disbursement is 0, the pending disbursement status is now set to "Reapplied". This setting prevents the disbursement from being selected for adjustment when the batch process is run again.
Disbursement Payment	
CC-57680	Added payment method detail fields to the Disbursement entity, which you can access using preupdate rules. Note that outgoing payments are still synchronized with the payment method fields on the Account, not on the Disbursement.
Charge/Invoice	
CC-58078	The Create New Invoice button on the Account Invoices screen is now functional. To create a new invoice, select an invoice stream and then click the button. You can modify the invoice and due dates for the invoice. You can also choose whether to create an ad hoc or a regular invoice. The default is ad hoc. BillingCenter treats ad hoc invoices differently from other invoices. For example, a held charge will not be placed on an ad hoc invoice. Instead, the item is placed on the next planned invoice in the invoice stream.
CC-58953	Fixed an issue where Invoice Bill Batch would throw an exception and fail to process an account if its billing plan's Fix Draft Day on was set to Last Business Day of Month.
Integration	

ID	Description
PL-12964	<p>Added a second version of the Document Assistant ActiveX control that Guidewire provides to manage many of the functions related to document manipulation. The default version uses a whitelist of file types that the Document Assistant control will open and display. The blacklist version provides a list of known potentially dangerous file types that it prevents from opening under any circumstances.</p> <p>For information on how to select the appropriate version of the Document Assistant, and for information on how to install and configure the Guidewire Document Assistant ActiveX control, see the System Administration Guide, topic <i>Configuring Guidewire Document Assistant</i>.</p> <p>WARNING The Guidewire Document Assistant, like all ActiveX controls, has known security vulnerabilities.</p>
Manageability-Platform	
PL-9368	Added a Download Comparison Zip File button to the Server Tools Info Pages → Data Distribution Information page if you select multiple data distributions. This provides the ability to merge multiple data distribution reports. This change also adds an additional column on the distribution reports index page that shows the origin distribution number
PL-10943	Modified the Server Tools Info Pages → Load Integrity Checks so that it no longer downloads duplicate entries. The process now respects the status of the Allow Non Admin References : drop-down option.
PL-11812	Added a new <i>action</i> attribute to the <code><tablestatistics></code> element in config.xml. By default, ClaimCenter on Oracle does not generate statistics on any table used for processing work items. ClaimCenter deletes any existing statistics on these tables whenever ClaimCenter updates statistics. You can override this behavior using the <i>action</i> attribute of the <code><tablestatistics></code> element. See the System Administration Guide, <i>Configuring Database Statistics</i> , for details.
PL-12143	Corrected an issue that caused poor performance on an Oracle database if you performed a linguistic search in which the Search strength was set to primary or the default application locale was Japanese or German.
PL-12806	Modified the Administration → Internal Tools → Database Table Info page so that it now includes database partition information.
PL-12854	Modified the ClaimCenter upgrader so that it now ignores the status of database constraints created by Oracle supplemental logging.
PL-12889	Corrected an issue in which the Work Queue Info page did not include the Throughout column in a download to a CSV file. This occurred primarily if there was a big disparity in the dates or the worker completed immediately.
PL-13048	Modified the Server Tools Info pages → Data Distribution page to include information on the number of updated rows and the total number of updates in HTML tables.
PL-13461	Modified the Server Tools Info pages → Data Distribution Information download logic so that it no longer generates data on all assignables by default. Formerly, a data distribution download generated large amounts of data on all assignables, even if you did not select the Assignable by date option. Now, you must select that option to view this information.
PL-13549	Corrected an issue that caused an <code>ArrayIndexOutOfBoundsException</code> if you performed a Server Tools Info pages → Data Distribution query on a very large database.
Payment Distribution	
CC-58929	Fixed an issue where, if an account had any Agency Bill policies with amounts outstanding, the automatic disbursement batch process would not create a disbursement. This impacted customers who mix agency and direct bill policies, as they could not create disbursement on those accounts. With this fix, accounts with Agency Bill policies can do disbursements.
Payment	
CC-58863	Fixed an issue where an exception occurred when the NewPayment batch tried to process undistributed reversed payments.
Payment Plan and Invoicing	
CC-58873	Optimized handling of payment plan modifiers so that if a billing instruction has a payment plan modifier, BillingCenter reuses a matching custom payment plan if one exists rather than creating multiple custom payment plans.

ID	Description
CC-58875	<p>Each payment plan now has optional overrides for use with each type of billing instruction. The payment plans can be edited in the user interface to specify which settings to override for each type of billing instruction. For example, you can create a payment plan that has a 10% down payment, but has an override of 0% down payment for charges that are created with a PolicyChange billing instruction.</p> <p>If a charge's billing instruction has any payment plan modifiers, they are applied to the payment plan *after* the plan is configured for the particular type of billing instruction.</p>
CC-58877	Fixed the issue where refactoring a payment plan with multiple charges and payment plan modifiers did not slice charges in a deterministic order.
CC-58878	Implemented new MatchPlannedInstallments payment plan modifier, which matches the planned installments related to a specified reference charge. This modifier replaces the MatchNumberOfPlannedInstallments modifier, which has been deleted.
CC-59079	<p>Changed the column PlcyBillingInstruction.ModifiedPaymentPlan to PaymentPlan. A batch process must be run after the SQL upgrade triggers to populate this column. After the SQL upgrade triggers finish, consistency checks will warn that some executed PlcyBillingInstructions have a null PaymentPlan. After the batch process completes, the consistency checks should pass.</p> <p>The batch process is BatchProcessTypeConstants.TC_UPGRADEBILLINGINSTRUCTIONPAYMENTPLAN "UpgradeBillingInstructionPaymentPlan"</p>

Policy Administration Integration

CC-59087	When a policy is renewed or rewritten (i.e., a new PolicyPeriod is created), BillingCenter takes the term number from the prior policy period, increments it by 1, and uses that for the term number for the new policy period. However, if for some reason a term number was "skipped" (e.g., due to a renewal that's not bound and is instead withdrawn), BillingCenter would not be aware of this and would simply use the incremented term number, which would then be out of sync with respect to what PolicyCenter has for the term number. To address this, the term number can now be modified (i.e., the setter is now scriptable) so that it can be set explicitly as part of sending the billing instruction. In this way, PolicyCenter can inform BillingCenter of the correct term number to use.
----------	--

Reporting

CC-58820	Account Balance report now maintains correct balance after reversals of money received, writeoffs, and disbursements.
----------	---

Studio

PL-13220	Improved Studio activation performance if using SCM systems that maintain a local repository (for example, SVN and CVS) and Studio is configured with proper repository mask in Tools Options.
PL-13509	Modified Studio to provide ability for time-sensitive diagnostic logging in the Studio console. You can enable this functionality by creating a studio.properties file in Other Resources → logging and populating it with the following text: # enable studio debug level logging log4j.category.Studio=DEBUG

Trouble Ticket

CC-58775	The TroubleTicketEscalation batch process has been converted to use the work queue mechanism.
----------	---

Web

PL-12262	Corrected an issue with Date input containing Japanese characters that sometimes caused the date text to not be fully visible in the text box. This caused the user to scroll the text to see the last part of the date.
PL-12842	Improved slow performance in Internet Explorer 7 that occurred if you moused over a page with a very long ListView and a Worksheet.
PL-13043	Modified the behavior of the Eras drop-down in the Japanese Imperial Calendar widget so that it displays only the most four recent eras (Meiji, Taicho, Showa, and Hisei).

ID	Description
Write Off	
CC-59058	Removed the ability to select a policy period as the target of a negative writeoff from the BillingCenter user interface.
CC-57527	Fixed an issue where an exception occurred when you attempted to edit an agency bill payment that was created after the policy's payment plan was changed.
CC-58696	When you set the disposition to "Writeoff" on a fully paid agency bill payment, you no longer receive an exception, and BillingCenter correctly saves your change.
CC-58717	Corrected an issue where BillingCenter allowed the execution of a billing instruction containing a charge based on a charge pattern that doesn't belong on that kind of billing instruction. BillingCenter now throws an illegal argument exception in this case.
CC-58780	Removed the NegativeChargeBilled, AgencyChargeBilled, and DBChargeBilled transaction subtypes. When an item is billed, a ChargeBilled transaction is issued regardless of the amount.
CC-58811	Account Balance report now maintains correct balance after (flat) policy cancellation.
CC-58968	Fixed an issue where BillingCenter would not persist changes to suspense items on an Agency Bill Payment if the change was made to a custom field.
	Modifying Suspense Items and Modifying Payment Items will now take extension fields into account when compared to the original items for determining whether to keep or discard themselves.
	Extension fields must be added to the Base Payment/Suspense Item in order for it to be considered (extensions added to the modifying item will continue to be ignored).
CC-59110	Fixed an issue where creating modifying entities (e.g., <code>ModifyingPaymentItem</code>), would not copy foreign key extension fields to the modifying item. BillingCenter now copies all "links" (i.e., foreign keys, edge foreign keys, and one-to-ones) when creating modifying entities. This fix does not copy extension arrays, as there are as-yet unknown performance implications that we need to investigate.

Known Issues and Limitations

Note: For maintenance releases, it is Guidewire policy to defer fixing configuration issues if they require merging files during the upgrade. Workarounds to many of these issues are listed in the following sections. The goal of this policy is to make upgrades as straightforward as possible.

This section describes known issues with this release of Guidewire BillingCenter.

Acrobat Sample document template does not allow extra fields (PL-1416)

Issue: Currently, the example Acrobat document template uses two fields, and this requires the same number of fields in `SampleAcrobat.pdf.descriptor` and `SampleAcrobat.pdf` files.

Workaround: To allow more fields in the template and documents created from it, you must update its descriptor file.

Server exception during startup (PL-8167)

Issue: During server startup, there can be a Tomcat server exception. However, after the exception, the server does start up. This is caused by the Tomcat application server attempting to either save or restore sessions using serialization. The session restore causes an exception because BillingCenter sessions contain objects which rely on the metadata being started up.

Workaround: None. The Tomcat application server does start up after the error.

Zone data with missing values cannot be loaded (PL-9123)

Issue: Zone data, which is used for associating postal codes with cities and regions, does not load if it is incomplete.

Status: Guidewire is aware of this issue.

Not Possible to Hot-Deploy on all Supported Application Servers (PL-9955)

Issue: It is not possible to hot-deploy on all supported application servers.

Workaround: If you use Tomcat, you are able to see the changes without requiring a restart of the application server. If you use an other application server, you need to perform a restart to see any configuration changes reflected.

Values in MIME-Type Drop-Down Are Not Localized (PL-10921)

Issue: BillingCenter does not localize the values in the MIME type drop-down list properly.

Workaround: If you want to localize MIME types, you can do so by modifying the definition in config.xml. Currently, BillingCenter does not support MIME type definitions in multiple languages.

Finder in Annotation Causes Throws Error (PL-10812)

Issue: Studio throws error and fails to do syntax error highlight, code completion, and similar methods, if there is a finder in an annotation. This occurs even if you are connected to the server.

Workaround: Remove the finder from the annotation.

Navigating Between Pages in Wizard Loses Data (PL-10920)

Issue: In a wizard pop-ups, it is possible to lose data if you navigate to another page in the wizard.

Workaround: Navigate out of the wizard and then back into it rather than changing steps directly in the wizard.

Studio Does Not Properly Report Verification Error (PL-10822)

Issue: Studio does not properly report a verification error if a typecode contains a trailing space.

Workaround: Remove trailing spaces from all type codes.

Double Quotes in Path Environment Variable Causes Problems (PL-10981)

Issue: BillingCenter does not run if there are double quotes in the PATH environment variable.

Workaround: Check the PATH variable and remove any double quotes.

Studio Debugger Does Not Handle Rule Breakpoints Properly (PL-10553)

Issue: If you set a breakpoint in a Gosu rule and attempt to step through the code using the Studio debugger, the debugger does not correctly display the code on the breakpoint step.

Status: Guidewire is aware of this issue.

Inability to Access Typecodes that Contain a Dash (PL-10532)

Issue: It is not possible to access a typekey code (typecode) that contains a dash using Gosu.

Workaround: Remove any dashes from the typekey codes.

Linguistic Search Performance on Oracle is Slow (PL-12143)

Issue: BillingCenter uses an Oracle Java Stored Function to perform linguistic searches. Such search queries can perform poorly. BillingCenter invokes the stored function on Oracle if either one of the following is true:

- The primary locale language is German or Japanese
- The `LinguisticSearchCollation` strength is set to `primary` in the default locale's `GWLocale` element in file `localization.xml`.

This is an issue for any linguistic search in which you set the `LinguisticSearchCollation` strength to `primary`. For example, if you are working in a language that commonly uses accents (such as French or Spanish), and you wish to perform accent-insensitive searches, then you typically set the `LinguisticSearchCollation` strength to `primary` for that language. If you do so, it is possible to experience slow performance on an Oracle database as you perform linguistic searches.

Workaround: Guidewire recommends that you set the `LinguisticSearchCollation` strength to `secondary` if applicable. (This is not applicable for Japanese and German.)

InetSoft Exception Failed Login Exception: Bad user name anonymous (PL-10148)

Issue: InetSoft generates a `Failed Login Exception: Bad user name anonymous` exception on server start-up. This occurs if you set the InetSoft logging level to `fine`.

Workaround: This error does not cause any issues. You can safely ignore the exception.

Report Still Visible After Permissions Disabled For Parent Folder (PL-10353)

Issue: An InetSoft report that is included in a permission set is still visible in the report tree even after its parent has been removed from the permission set.

Workaround: You must remove individual reports from permission sets instead of maintaining the reports at the report folder level.

ToolbarButton Widget Does Not Render Correctly (PL-10742)

Issue: BillingCenter incorrectly renders a `ToolbarButton` widget if the associated `AddMenuItemIterator` is empty.

Workaround: Modify your configuration to disable the menu button if the button contains no menu item.

Must Name Row Iterator Element with Capital Letter (PL-10795)

Issue: The Document search results page (`DocumentLV.pcf`) throws a `NullPointerException` if the row iterator element name is `document` (with a lower case d).

Workaround: You must use `Document` (with a capital D) instead.

Large Number of Classes in plugins\shared\lib Causes Studio to Fail to Start (PL-11026)

Issue: If you attempt to load multiple JAR files that contain many thousands of classes, Studio fails to start and throws an `OutOfMemory` error.

Workaround: Increase `MaxPermSize` to something over `128m` (the default) so that Studio has enough memory to build the type information database.

ANT Command `build-war` Hangs Indefinitely (PL-11045)

Issue: It is possible for the `build-war` ANT command to hang indefinitely. This issue is only reproducible on certain environments.

Workaround: Set the `-noinput` parameter before calling the `build-war` ANT command. For example, instead of `gwcc.bat build-war`, do the following:

```
gwbc.bat -noinput build-war
```

[Missing Display Keys Cause Server to Fail to Start \(PL-11764\)](#)

Issue: Missing display keys defined in the application default locale can cause the server to fail to start.

Workaround: Within Studio, run the **Verify Resources** utility before attempting to start the application server. BillingCenter bases the type system for display keys on those defined in the default application locale. If a display key in the default application locale is missing and a rule references the missing display key, then BillingCenter cannot compile the rule. Running the resource verification utility can catch these types of missing display keys.

[BillingCenter Not Generating `soap.local.entity` Correctly \(PL-7560\)](#)

Issue: BillingCenter is not generating the `soap.local.entity` for a class used by web services.

Workaround: To ensure that BillingCenter generates the `soap.local.entity` correctly for `FieldValue` classes (which are Java types), subtype `FieldValue` into `GosuFieldValue`, then further subtype that class.

[Incorrect additivity statements in the default `logging.properties` file \(CC-50615\)](#)

Issue: Some additivity statements in the `logging.properties` file are incorrect and therefore, ineffective. An additivity statement attempts to keep the category from sending its log messages up to its parent, preventing duplicate log messages across files.

Workaround: In the `logging.properties` file (located at `modules/cc/config/logging`), change:

```
log4j.additivity.PluginsLog=false  
to  
log4j.additivity.Plugin=false
```

[BillingCenter login remains active in new browser window \(CC-4152\)](#)

Issue: When you log into BillingCenter in a browser window, then you close that window and open a new one, in some instances you can access BillingCenter pages without needing to log in again.

This is an issue with some versions of Microsoft Internet Explorer. When multiple browser windows are running in the same process, they can share session information, and this allows them to share a single login with BillingCenter. Early versions of Internet Explorer provided a **Launch browser windows in a separate process** option, which, if enabled, would avoid this situation. Some versions of Internet Explorer do not allow you to set this option, and enable or disable it automatically. You can review the relevant Microsoft Knowledge Base Article 240928. However, you may experience a situation in which Internet Explorer does not correctly enable this option, especially if you manually disabled it in an earlier version and then later upgraded. In this case, the multiple browser windows are run in the same process, exhibiting the undesired behavior.

Workaround: The simplest solution is to always click **Logout** at the top of the BillingCenter window before closing your browser.

You can also take steps to manually enable the Internet Explorer option. This requires you to edit the Windows registry on the affected computer.

WARNING Editing the registry is a risky procedure, and any improper changes may result in permanent data loss and require Microsoft Windows to be completely reinstalled on that computer. To make this change, create or edit the

`[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\BrowseNewProcess]` registry entry. Set the data type to `REG_SZ`, and then set the data value to `yes`.

WebSphere timeout during initial startup (CC-4904)

Issue: The initial creation of the BillingCenter database may take longer than the default WebSphere startup timeouts will allow, and WebSphere may abort the startup before the database upgrade has completed. In this situation, the `Default_Server_stderr.log` file shows an interruption in the `Applying upgrade step #### of #####` log messages.

Workaround: Increase the Ping Timeout and Ping Initial Timeout values for WebSphere. These can be found in the WSADMIN Console. Click **Nodes**, drill down to your application server, click the **Advanced** tab, and increase the values.

BillingCenter displays many warnings when regenerating Javadoc (CC-55564)

Issue: This issue occurs when running the `regen-java-api` and `regen-soap-api` command.

Status: Guidewire is aware of this issue.

On AIX without an X11 server, BillingCenter may report that an X11 resource is missing (CC-5889)

Issue: BillingCenter does not require an X11 server; however other components of the BillingCenter environment (such as Java) may inadvertently have this requirement. If so, BillingCenter may report that an X11 resource is missing when it is starting up.

Workaround: Install an X11 server, or contact Guidewire support to determine if the dependency can be removed.

Port conflict when starting BillingCenter with JMX RMI adapter enabled and Microsoft Outlook running (CC-6964)

Issue: When Microsoft Outlook starts, it chooses several ports to use for its connection to an Exchange server. In some instances, Outlook may use port 1099, which is a port that the BillingCenter JMX RMI adapter can use, *if enabled*. The BillingCenter server then reports a port conflict during startup when it tries to use that port for the RMI adapter.

Workaround: Close Outlook to free the port, and then start the BillingCenter server again. You can also change the port that the RMI adapter uses by editing the `JMXManagementPlugin` through Guidewire Studio.

Studio PCF Editor treats commented-out property values as errors (PL-4582)

Issue: If you attempt to comment out a property value in the Studio PCF editor, Studio treats this as an error.

Status: Guidewire is aware of this issue.

Incomplete implementation of non-English locale-specific database search and sort (PL-8108)

Issue: Guidewire has begun implementation of database searching and sorting appropriate to non-English locales. This release of BillingCenter contains an incomplete implementation of this work. The changes will not affect customers implementing case-insensitive searches using standard English sort criteria.

As part of this work, the current release renames the `supportsCaseInsensitiveSearch` attribute on column to `supportsLinguisticSearch`.

Status: Guidewire is aware of this issue.

Server exception during startup (PL-8167)

Issue: During server startup, there can be a Tomcat server exception. However, after the exception, the server does start up. This is caused by the Tomcat application server attempting to either save or restore sessions using serialization. The session restore causes an exception because BillingCenter sessions contain objects which rely on the metadata being started up.

Workaround: None. The Tomcat application server does start up after the error.

[Zone data with missing values cannot be loaded \(PL-9123\)](#)

Issue: Zone data, which is used for associating postal codes with cities and regions, does not load if it is incomplete.

Status: Guidewire is aware of this issue.

[H2 development database creates LONGTEXT fields as VARCHAR\(65000\), instead of as CLOB as in Oracle and SQL Server \(PL-9314\)](#)

Issue: Guidewire creates schema fields defined as LONGTEXT as LONGVARCHAR columns, instead as CLOB as in the Oracle or SQL Server databases. In addition to the inconsistency involved, this mandates a field length of 65,000 characters or less in LONGTEXT columns in the H2 database.

Workaround: Define a LONGTEXT field as a VARCHAR(...) with the required size, for example, as VARCHAR(120000).

[Report Folders Do Not Display If Empty \(PL-1281\)](#)

Issue: Guidewire BillingCenter does not display a report folder if that folder is empty.

Workaround: Place an InetSoft replet inside the folder. You can mark the replet as *not visible* within the InetSoft Enterprise Manager. This ensures that the parent folder displays in BillingCenter even if the folder is empty.

[Report Folders Do Not Display If Empty \(PL-1281\)](#)

Issue: BillingCenter does not display a report folder if that folder is empty.

Workaround: Place an InetSoft replet inside the folder. You can mark the replet as *not visible* within the InetSoft Enterprise Manager. This ensures that the parent folder displays in BillingCenter even if the folder is empty.

[Not Possible to Hot-Deploy on all Supported Application Servers \(PL-9955\)](#)

Issue: It is not possible to hot-deploy on all supported application servers.

Workaround: If you use Tomcat, you are able to see the changes without requiring a restart of the application server. If you use an other application server, you need to perform a restart to see any configuration changes reflected.

[Values in MIME-Type Drop-Down Are Not Localized \(PL-10921\)](#)

Issue: BillingCenter does not localize the values in the **MIME type** drop-down list properly.

Workaround: If you wish to localize MIME types, you can do so by modifying the definition in `config.xml`. Currently, BillingCenter does not support MIME type definitions in multiple languages.

[Finder in Annotation Causes Throws Error \(PL-10812\)](#)

Issue: Studio throws error and fails to do syntax error highlight, code completion, and similar functions, if there is a finder in an annotation. This occurs even if you are connected to the server.

Workaround: Remove the finder from the annotation.

[Navigating Between Pages in Wizard Loses Data \(PL-10920\)](#)

Issue: In a wizard pop-ups, it is possible to lose data if you navigate to another page in the wizard.

Workaround: Navigate out of the wizard and then back into it rather than changing steps directly in the wizard.

Studio Does Not Properly Report Verification Error (PL-10822)

Issue: Studio does not properly report a verification error if a typecode contains a trailing space.

Workaround: Remove trailing spaces from all type codes.

Double Quotes in Path Environment Variable Causes Problems (PL-10981)

Issue: Guidewire BillingCenter does not run if there are double quotes in the PATH environment variable.

Workaround: Check the PATH variable and remove any double quotes.

Studio Debugger Does Not Handle Rule Breakpoints Properly (PL-10553)

Issue: If you set a breakpoint in a Gosu rule and attempt to step through the code using the Studio debugger, the debugger does not correctly display the code on the breakpoint step.

Status: Guidewire is aware of this issue.

Incorrect totals in the policy payment schedule occurred on agency bill statements after a change was made to one of two producers on a policy (CC-56925)

Issue: The producer statement does not show the correct dollar amount after a change is made to one of the policy's payment plans on a policy with two producers after the statement was billed and partially paid.

Status: Guidewire is aware of this issue.

BillingCenter did not process policy changes done after the policy expiration date (CC-56852)

Issue: During BillingCenter - Policy Center integration, BillingCenter does not process policy changes made after the policy expiration date. An illegal state exception displays on both the BillingCenter and PolicyCenter consoles.

Status: Guidewire is aware of this issue.

Moving a Billed invoice item should negate the direct bill commission paid, not the agency bill commission paid (CC-56656)

Issue: If an item is moved from a Billed invoice to a Planned invoice, an earnings transaction is created for the Payable amount on the commission. However, if the commission was earned and paid, it is not being negated.

Status: Guidewire is aware of this issue.

After a reversal, the distributed payment is still being reserved for the premium payment (CC-56538)

Issue: The disbursement amount is not fully moved to disbursement and does not show up on the Disbursement page although an entire payment has been reversed and the associated charge had been assigned to another account.

Workaround: This is a new feature that is not fully implemented. Guidewire is aware of this issue.

Errors occur in the totals for the producer's rate on the Transactions screen (CC-55835)

Issue: A difference occurs in the totals shown on the Transactions screen for a producer when selecting Reserves and then choosing either the Last 30 Days or Show All filter.

Workaround: Guidewire is aware of this issue.

New collateral charges and invoice amounts are being generated based on previously paid collateral amounts, not the previously billed or planned amounts (CC-55209)

Issue: New collateral charges and invoice amounts are being generated based on the amount of collateral previously paid and do not account for amounts of collateral previously Billed or Planned (not yet billed). This results in an over billing of the collateral amount on the invoices.

Workaround: Guidewire is aware of this issue.

No commission reserve and onset transfer transaction was created when a direct bill item was reassigned to an agency bill producer (CC-54923)

Issue: When a direct bill policy has no specified primary producer and one of the policy's item is reassigned to an agency bill producer, no commission reserve and onset transfer transaction created. This was because there is no initial reserve.

Workaround: Guidewire is aware of this issue.

Billing instructions for Issuance and Policy Change must work with premium and other charges (Premium Reporting Only) (CC-52032)

Issue: This issue is for Premium Reporting policy periods only. Users are note allowed to enter premium charges as part of an issuance or policy change and process them as any other policy period. This issue applies to all billing instructions types.

Workaround: Guidewire is aware of this issue.

Automatic distribution to policies not working correctly (CC-51511)

Issue: Automatic distribution to policies on Account → Disbursement screen does not distribute the payment correctly.

Workaround: Guidewire is aware of this issue.

Problems occurred with the use of negative collateral charges (CC-51312)

Issue: When issuing a negative collateral charge, the amount gets subtracted from a Charge Due T-account and a payer's Unapplied T-account. This causes an imbalance between collateral reserve and collateral held when issuing a negative charge.

Workaround: Guidewire is aware of this issue.

Cannot undistribute direct bill payment (CC-58764, CC-58764)

Issue: The base configuration does not allow a direct bill payment to be undistributed.

Workaround: Use the configuration to create an Unapply button on the Payment List View. This button should remove the original DirectBillMoneyRcvd, reverse all associated transactions, and then create a brand new DBMoneyRcvd with the same attributes as the original.

No API to remove invoice items from DirectBillPaymentView (CC-58397)

Issue: There is no way to programmatically remove an Invoice item (or invoice item array) from a distribution.

Status: Guidewire is aware of this issue.

Invoice PreUpdate rule is bypassed when a new invoice created for Collateral charge (CC-58254)

Issue: Automatic distribution to policies on Account → Disbursement screen does not distribute the payment correctly.

Status: Guidewire is aware of this issue.

Reversing Write-Off Commissions and Write-Off Commission Recovery in the UI (CC-58655)

Issue: The UI does not provide a way reverse Write-Off Commissions and Write-Off Commission Recovery.

Workaround: Clone the Reverse button under the Payments tab and reverse CmsnPayableReduction and other transactions using the following action: CurrentLocation.startEditing(); producerTransaction.reverse(); and CurrentLocation.commit().

Use of StringBuilder Class prevents debugging of web service (CC-58533)

Issue: In a web service, if you have a method with a parameter of type X and inside the web service method you reference a Gosu class of type Y. And, Gosu class Y has a `toString()` method (overriding the default) that uses a `java.lang.StringBuilder` class. If you set a breakpoint in the web service method and try to step through the method, then you will get errors.

Status: Guidewire is aware of this issue.

Should not be able to edit Payment Plan after it has been used by a policy (CC-57758)

Issue: In the base configuration, it is possible to edit a Payment Plan even after it has been used by a policy. Only the expiration date should be updateable.

Workaround: Configure the payment plan screen to only allow the expiration date to be updated.

Account Evaluation is not counting account level delinquencies (CC-55938)

Issue: The count for number of delinquencies on Account\Evaluation screen doesn't factor account level delinquencies.

Workaround: Use the Gosu finder to identify the account level delinquency processes and add it to the BillingCenter value.

Duplicate payment plan database constraint on starting an integrated environment and sending cancelPolicy message (CC-56972)

Issue: BillingCenter does not process policy changes done after the policy expiration date. Illegal state exception is received in console of both BillingCenter and PolicyCenter.

Status: Guidewire is aware of this issue.

BillingCenter does not process policy changes done after the policy expiration date (CC-56852)

Issue: BillingCenter does not process policy changes done after the policy expiration date. Illegal state exception is received in console of both BC and PolicyCenter.

Status: Guidewire is aware of this issue.

Amount Distributed on a payment does not reflect transfers.(CC-58782)

Issue: The Amount Distributed on a payment does not reflect transfers. This means that the Account : Unapplied is adjusted, but the full payment amount is still available for distribution on the original account.

Workaround: Make the following amendments to the `allocatePayment` method of the `DirectBillPayment.gs` plugin. Add the following lines to adjust the amount to allocate before you call out to the allocation strategy:

```
allocatePayment( DirectBillPayment, Amount)
if ( payment.BaseMoneyReceived.Executed) {
    amount = BigDecimalUtil.min(payment.Account.RawUnappliedAmount, amount)
} else {
    amount = BigDecimalUtil.min(
        payment.Account.RawUnappliedAmount.add(payment.BaseMoneyReceived.Amount),
        amount)
}
```

The same issue will happen when calling the following plugin methods:

- `allocateCollateral`
- `allocatePolicyPeriod`
- `allocateCredits`.

You can add the same lines of code to these plugin methods to correct the problem that occurs if the payment you are allocating has more money in it than what is actually executed against Account : Unapplied.

[Tab widgets do not work with Microsoft Windows Server 2003 enhanced security enabled \(PL-9813\)](#)

Issue: If you have the Windows Server 2003 component called IE Enhanced Security Feature installed, tabs may be unresponsive in the user interface.

Workaround: Uninstall this component from the Windows 2003 Server.

[Cache summary statistics rely on server ID being unique \(PL-11239\)](#)

Issue: If you use the same server ID for multiple servers in a cluster, it can create the following problems:

- Different servers in a cluster deleting or changing each others results
- Deadlocks at the database level

Workaround: By default, the server ID defaults to the hostname of the machine running the instance. This works well in many configurations, but causes problems if multiple server instances are running on a single machine. Set the `serverid` system property in the `config.xml` file on the local server host to a unique value in the cluster. If it does not currently exist, then you need to add a `<registry>` element to the `config.xml` file on the local server host.

You can also start the server with the following JVM option: `gw.bc.serverid=????`

See *Defining the Application Server Environment* in the System Administration Guide for details on both of these options.

[Countries configured in zone-config.xml still generate a warning during regen-dictionary even when zone data is loaded for all of these countries \(PL-11947\)](#)

Issue: Countries configured in `zone-config.xml` still generate a warning during `regen-dictionary` even when zone data is loaded for all of these countries.

Workaround: Warning message is created in error and can safely be ignored.

[The user interface cannot handle starting multiple instances of a batch process \(PL-12372\)](#)

Issue: The user interface cannot handle multiple instances of a batch process.

Workaround: If multiple instances of a batch process need to be executed, they must be started from the command line. Also, you should ensure that the `BatchProcess.isExclusive()` method returns `false` to allow multiple instances to be run simultaneously.

[First time you click on the arrow of the typekey input, the drop-down menu will not open \(PL-10134\)](#)

Issue: The drop-down menu does not open on the first click of the arrow on a typekey input. Instead, the help text opens.

Workaround: Turn off help text on focus by setting `InputHelpTextOnFocus` to `false` in the `config.xml` file. By doing that, the help text shows only if you mouse over the input and will not interfere with opening a drop down menu.

[There is a length limitation on entity localization table names \(PL-13360\)](#)

Issue: There is a length limitation on entity localization table names.

Workaround: Ensure that localization `tableName` property specified in the entity extension file is less than 16 characters. The error message generated if the localization table name exceeds the maximum length indicates that 18 characters are allowed, but that does not account for two additional characters added by the application.

[Type system refresh after PCF page title change does not update corresponding menu label \(PL-13057\)](#)

Issue: The type system refresh after the PCF page title change does not update corresponding menu label.

Workaround: After updating a page title, the server restart must be done to refresh menu labels and avoid null pointer exceptions due to stale reference.

[The GX model generated XSD cannot be parsed by JAXB \(PL-13598\)](#)

Issue: The GX model generated XSD cannot be parsed by JAXB.

Workaround: Add JAXB annotation elements to the XSD to specify the necessary metadata (such as class names) for JAXB to generate the java class files. Contact Guidewire Support for an example XSD annotated in this way.

[US-Locations.txt file with the US geodata from GreatData has special characters that cause validation problems with United States Postal Service \(USPS\) data \(PL-13384\)](#)

Issue: The `US-Locations.txt` file contains information that does not conform to United States Postal Service (USPS) standards for bulk mailings.

Workaround: The provided `US-Locations.txt` file is intended only for use in geocoding to identify addresses for a location. You can process the `US-Locations.txt` file to conform to your particular address standards, and then import that version of the file instead.

[GX models that reference virtual fields and enhancements throw null pointers if null \(PL-13560\)](#)

Issue: The GX models that reference virtual fields and enhancements throw null pointers when they are null.

Workaround: Ensure that null checks and error handling is included so that if referenced virtual fields or enhancements are null, then there will not be a null pointer exception.

[Sending email with file attachment with unicode filename is not correctly handed over to the mail server \(PL-13582\)](#)

Issue: Sending email with file attachment with unicode filename is not sent to the mail server correctly.

Workaround: Use Latin characters for file names on attached files.

[JavaToolkit.gs has incorrectly hard coded memory which results in failed regen-java-api ant task \(PL-13663\)](#)

Issue: The JavaToolkit.gs has hard coded memory which can result in failed regen-java-api ant tasks.

Workaround: Increase the size of the max heap setting on line 161 of JavaToolkit.gs in the Ant module. The shipped value is 512.

[Copy and paste does not work from phone fields \(PL-13792\)](#)

Issue: The copy and paste does not work for phone fields on a contact.

Workaround: Save the contact with the newly entered phone information before attempting to either copy or paste the value into another field.

[You cannot make a field from a delegate a localized column \(PL-13761\)](#)

Issue: You cannot make a field from a delegate a localized column.

Workaround: Move the column to be localized off the delegate and onto each of the implementing entities. Then, in order to make the column appear as though it exists on the delegate, define an enhancement property on the delegate, that *delegates* to the appropriate column, depending on the implementing entity.

[Debugger break point for the rule engine does not work if first the line is a comment or a blank line \(PL-12912\)](#)

Issue: The debugger break point for the rule engine does not work if the first line is a comment or a blank line.

Workaround: Ensure that the first line is not blank and does not contain a comment. Add comments after the first line.



chapter 34

Guidewire BillingCenter 3.0.2 Release Notes

Release 3.0.2.29

Copyright © 2001-2010 Guidewire Software, Inc. All rights reserved. Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, and the Guidewire logo are registered trademarks, and Guidewire Insurance Suite, Guidewire BillingCenter, Guidewire ContactCenter, and Guidewire Studio are trademarks of Guidewire Software, Inc. All other product names are the trademarks of their respective companies.

This product includes information that is proprietary to Insurance Services Office, Inc (ISO). Where ISO participation is a prerequisite for use of the ISO product, use of the ISO product is limited to those jurisdictions and for those lines of insurance and services for which such customer is licensed by ISO.

This documentation is published as Guidewire Confidential. The contents of this documentation, including product architecture details and APIs, are considered confidential and are fully protected by customer licensing confidentiality agreements and signed Non-Disclosure Agreements (NDAs).

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire BillingCenter.

Version Number

- This release of Guidewire BillingCenter is 3.0.2.29
- For Standard Reporting, it supports InetSoft StyleReport Enterprise Edition 10.1, 20100113.

Support

This document includes important information about the current release of Guidewire BillingCenter. For assistance with this software release, contact Guidewire Customer Support. Contact support on the Web at <http://portal.guidewire.com>, by email at support@guidewire.com, or by phone at +1-650-356-4955.

Installing This Release

For installation information, see the *BillingCenter Installation Guide*.

InetSoft Evaluation

You can evaluate InetSoft Style Report Enterprise Edition Version 10.1 using the following license key:

L000-798-ERX-0000BF00100001F-F2AFC9DD34ED

If you want to use Inetsoft and the BillingCenter reporting module in a production environment, contact Guidewire Customer Support to request a full license key. For more information, refer to the *BillingCenter Reporting Guide* documentation.

WARNING The build number of the reporting install files must match the build number of the Guidewire BillingCenter installation. If the build numbers do not match, you will have problems with your reporting installation.

Changes in this Release

This section describes the product changes in this release.

- Base PCF File Changes

Base PCF File Changes

All links below require the `readme_files` directory on your local disk.

BillingCenter release 3.0.1 to 3.0.2

- To view a report of the changes in the base PCF files in the `modules/bc` directory, [click here](#).
- To view a report of the changes in the base PCF files in the `modules/platform` directory, [click here](#).

General Issues and Improvements

The following are the improvements added to and the general issues corrected in this release:

Issue ID	Description
Account	
CC-58146	<p>The following changes have been made to the billing information on the Account Summary screen:</p> <ul style="list-style-type: none"> Renamed Payment Info to Billing Info Added Fix Bill Date/DueDate? field for specifying bill date or due date invoicing Added invoice stream names to invoice date fields (for example, Monthly -Day of Month) Exposed Twice a Month - Days of Month fields
CC-58204	<p>Prior to this fix, four fields on the Account Summary screen were always displayed as readonly / non editable. Now the same four fields are editable whenever the Account Summary screen is in edit mode. The fields are:</p> <ul style="list-style-type: none"> Fix Bill Date / Due Date Monthly - Day Of Month Weekly - Day Of Week Every Other Week - Anchor Date
CC-58247	<p>BillingCenter now supports bill date and due date invoicing for all four periodicities (monthly, twice a month, weekly, every other week). To accomplish this, the DueDayOfMonth and FixDueDayOfMonth columns were removed, and the DueDateBilling column was added. Due date billing is now done by checking the due date billing bit, and all monthly invoicing is calculated from the InvoiceDayOfMonth.</p> <p>AccountBuilder has been modified to specify whether dates are to be set using a fixed bill date or fixed due date.</p> <ul style="list-style-type: none"> withFixedBillDateOfMonth(i) has been removed from the account builder. Instead use .withInvoiceDayOfMonth(i) and asBillDateBilling() withFixedDueDateOfMonth(i) has been removed from the account builder. Instead use asDueDateBilling().withInvoiceDayOfMonth(i)
CC-58581	<p>The unit of work for the Invoice and Invoice Due batch processes has changed from invoice to Account and from statement to producer. In both cases, the invoices or statements are processed in order of event date. The number of work items processed correlates to the number of accounts or producers that had invoices or statements processed.</p>
CC-58641	<p>For a policy cancellation, the negative items on the canceled policy are used to pay off the only the positive items on that policy period. The remaining amount of credit will remain unused until it is processed by some other distribution strategy.</p>
CC-57487	<p>The PolicyPeriod : Unapplied T-account has been removed. Its functionality is now replaced by Direct Bill Suspense Items.</p>
Administration	
CC-58700	<p>Previously, new records were added in the middle of the authority limits file (authority-limits.csv). This caused a shift in the public ID values for the records that come after the newly inserted record. Data is now added to the end of the file to maintain existing hard references to public IDs in the customer configuration.</p>
Agency Bill	

Issue ID	Description
CC-57180	<p>Previously, BillingCenter allowed only one linear stream of statements within a single producer. There are cases that are not addressable in this structure, such as:</p> <ul style="list-style-type: none"> • Statements with due dates before the normal due date, such as for a large premium down payment or special charge • Separate statements for different types of items (such as commercial lines, personal lines, or separation by producer code) • Non monthly periodicities <p>For agency bill, BillingCenter now supports: Multiple invoice streams and Ad hoc statements (that is, statements off the normal cycle)</p> <ul style="list-style-type: none"> • You can issue a promise and/or payment to an ad hoc statement just like normal statement • Both snapshots and carry forward work correctly for ad hoc statements • Invoice stream filter added to agency bill Invoice screen • Agency bill cycles (or statements) with arbitrary start, end, and due dates at time of creation • Co-existence of more than one agency bill statement for a time period with no requirement to have the same start/end/due dates • API for creation of additional statements • Separate line items for agency bill cycles for each statement • Assignment (manual and automated) to and from out-of-cycle statements • Current item placement code can be redirected towards an out-of-cycle statement <p>In the base application, no process will result in items being placed or represented on these additional out-of-cycle statements.</p>
CC-58333	Commission % column on the Agency Bill Payment wizard now shows the correct value. Previously, it rounded to no decimal places.
CC-58492	The Add Items popup that is available when you modify an agency bill payment can now find producer items where there is still some gross to be collected or some commission to be paid.
CC-58527	Fixed an issue where negative charges created at the time of canceling an agency bill policy caused a negative ChargePaid transaction to be created.
CC-58536	The event date for agency bill statements now pays attention to the cycle close day of month logic so that it can be set to nextBusinessDay, firstBusinessDay, and so on. Previously, this date was always set to the cycle close day of month.
CC-58707	Fixed an issue where the subtotals for the Agency Payment wizard payment items were being calculated using the wrong fields.
Billing Plan	
CC-58124	When an account's Billing Information changes (for example, the Monthly - Day of Month field), the changes are now propagated to existing invoice streams.
Cancellation Process	
CC-58434	<p>The legacy InvoiceMatch/InvoiceMatcher have been removed from the system and no longer are used to customize how collapsed invoice items are placed when a PolicyPeriod is canceled. Instead, use these two new methods on the IPolicyPeriod plugin to prepare the collapsed invoice items for placement:</p> <ul style="list-style-type: none"> • <code>InvoiceItemPayers getCustomPayersForCancellationCollapsedInvoiceItems(InvoiceItemPayers defaultInvoiceItemPayers)</code> • <code>void updateCancellationCollapsedInvoiceItemsBeforePlacement(List<InvoiceItem> invoiceItems, Date cancellationCollapseDate)</code> <p>Use the InvoiceAssembler plugin to control exactly how to place collapsed invoice items onto cancellation invoices.</p> <p>As part of this change, BillingCenter no longer has the <code>AllowCancellationInvoiceItemsOnBilledAndDueInvoices</code> configuration parameter. In previous versions of BillingCenter, that boolean parameter controlled how to invoice the invoice items from charges in the Cancellation Billing Instruction. The parameter is now gone. To control how BillingCenter places invoice items from charges in a Cancellation Billing Instruction, put that logic into your implementation of the InvoiceAssembler plugin.</p>
Charge/Charge Group	
CC-58713	BillingCenter now provides the ability holds negative charges. For Direct Bill, all Negative Items will be carried forward regardless of Paid Status. For Agency Bill, negative items obey the same rules as positive items in that they will be carried forward as long as there are no payments against them.
Charge/Invoice	

Issue ID	Description
CC-56840	Removed the Edit Next Invoice Date button in the Account → Invoices list view. With the introduction of invoice streams, this button is not relevant on this screen. The functionality was previously moved to the Change Invoice Dates button on the Invoice card, which is shown when you select a planned invoice.
CC-57450	The Invoice Items list view on the Account → Charges screen displays the item payer (which could be an account or producer). For new items on the account, the field was incorrectly showing the producer name. It now shows the account number.
CC-57644	The InvoiceAssembler plugin method <code>getCustomInvoiceItemPlacements</code> now takes an additional parameter, which is an InvoiceAssemblerContext typecode. This object represents information about why BillingCenter called the invoice assembler. For example, if it was an agency plan change, it is the typecode <code>TC_AGENCYPLANCHANGE</code> .
CC-57801	On the Account → Invoices screen, a Product column (<code>Policy.LOBCode</code>) has been added to the aggregation for Policies, Charge Groups, and Charges. The Product column is hidden for Contexts and Categories aggregations. This is because these two aggregations can represent multiple policy periods (which could have different values for <code>Policy.LOBCode</code>).
CC-57906	Sometimes it is not possible to invoice any installments before the invoice blackout period. For example, this happens if the policy period is very short. In that case, the whole policy amount is now charged in one invoice item as a down payment (if a down payment exists in the payment plan) or as a single installment for the full amount (if no down payment exists). The date of the down payment or installment is determined by the Down Payment or First Installment settings on the payment plan. The single down payment or installment item is placed on the earliest invoice with an event date on or after the item's date.
CC-57945	A dropdown filter to filter by invoice stream has been added to the Account → Invoices and Producer → Agency Bill Cycles screens.
CC-57976	DirectBillInvoiceMatch has been removed. The InvoiceAssembler plugin performs the DirectBillInvoiceMatch functionality.
CC-58142	Due date invoicing now works with all billing periodicities, not just monthly. The <code>DueDateOfMonth</code> and <code>FixedDueDateOfMonth</code> fields have been removed from the account. Now monthly invoicing just uses the <code>InvoiceDayOfMonth</code> . A bit field (<code>DueDateBilling</code>) indicates whether the invoicing dates specify the invoice bill date or due date.
CC-58158	If account is set up with due date invoicing and the non-responsive lead time on the billing plan is different from the lead time, changing the account payment method from responsive to non-responsive and then changing payment plan for a policy period resulted in wrong invoices dates. Now changing the account's payment method removes all existing planned invoices and puts their items onto new invoices that use the new lead time.
CC-58316	For a short term policy period, the calculation of due date minus lead time can result in an invoice date that is before the current date. To avoid an error in this case, BillingCenter will bill at the next due date minus the lead time even if the resulting date falls after the policy period ends.
CC-58435	The <code>ChargeReversed</code> class that was used by the <code>InvoiceMatch</code> class is now removed. It helped place reversal invoice items that are generated when a user reverses a charge. The base implementation behavior does not change at all from this change. However if you want to customize the placement of invoice items when reversing a charge, configure the InvoiceAssembler plugin logic to handle this case.
CC-58604	Fixed an issue where if you searched for an invoice item that was not included in a payment, a null point exception was thrown when the invoice item was not found.
CC-58642	Switching a policy period's payment plan caused an exception if the policy period had two charges, each of which were added using payment plan modifiers. This problem is now fixed.
CC-58661	Offsets can be placed on billed invoices for direct bill (and for agency bill if the agency bill plan allows it). Before this fix, the offset was always placed on the earliest invoice after the event date of the invoice item. It was possible that the offset item was placed on a different invoice than the original item. The fixed behavior is to place the offset on the same invoice as the original item if it does not conflict with the setting for whether or not to allow placing offsets on billed invoices.
CC-58665	The offset for a billed item will be placed on the same invoice as the original item when it does not conflict with the setting allowing placing offsets on billed invoices.

Issue ID	Description
CC-58692	<p>You can use custom periodicities in payment plans and invoice streams. As part of this change:</p> <ul style="list-style-type: none"> The plugin interface <code>IInvoiceStream</code> changed. The <code>getInvoiceStreamFor(InvoicePayer, Charge, PaymentPlan, InvoiceStream)</code> methods are now gone and replaced with <code>getInvoiceStreamPeriodicityFor(InvoicePayer, Charge, Periodicity)</code>, <code>getExistingMatchingInvoiceStream(InvoicePayer, Charge, Periodicity)</code>, <code>customizeNewInvoiceStream(InvoicePayer, Charge, InvoiceStream)</code>, and <code>customizeInvoiceStreamAfterChangeToInvoicingDates(InvoiceStream)</code>. The methods <code>PaymentPlan.createInvoiceItems(Charge)</code> and <code>PaymentPlan.recreateInvoiceItems(Charge, BigDecimal)</code> are now deprecated. Use <code>Charge.createInvoiceItems()</code> and <code>Charge.recreateInvoiceItems(BigDecimal)</code> instead. The method <code>InvoicePayer.getInvoiceStreamFor(Charge, PaymentPlan)</code> is now deprecated. Use <code>Charge.getInvoiceStream()</code> or <code>InvoiceStreamFactory.getInvoiceStreamFor(InvoicePayer, Charge)</code> instead.
Commission/Commission Plan	
CC-56395	<p>The <code>ProducerCashToPayableTransfer</code> entity has been added. You can use the entity to issue a transaction that will debit the <code>CommissionsPayable</code> T-account of one producer and credit the <code>Cash</code> T-account of another producer.</p> <p>There are two scriptable methods on <code>ProducerCashToPayableTransfer</code>: <code>execute()</code> (which executes the transfer transaction) and <code>reverse()</code> (which reverses the transfer transaction).</p> <p>The <code>ProducerCashToPayableTransfer</code> entity has been associated to a new authority limit that is named <code>producercashtopayabletransfer</code>. Whenever an attempt is made to execute a <code>ProducerCashToPayableTransfer</code>, that authority limit is checked. Additionally, a new permission has been added to the system named <code>prodrevcashtopayablexfer</code>. This permission is checked whenever an attempt is made to reverse a <code>ProducerCashToPayableTransfer</code>. So, if a user does not have that permission, they will not be allowed to reverse a <code>ProducerCashToPayableTransfer</code>.</p>
CC-58051	<p>There was a problem that when a commission writeoff on a billed invoice was reversed (for example, when the policy changed from agency bill to direct bill) a reversal was created but an onset item was <i>not</i> created. Furthermore, the commission writeoff denormalized field on the invoice item was not updated correctly.</p> <p>This was changed so that the writeoff transaction is not reversed. The transaction, as well as the <code>CommissionRemainder</code> wrapper entity, are now both updated to point to the new onset invoice item, and the relevant denormalized fields are now set correctly.</p>
CC-58570	<p>Previously, there was no authority limit checking when a user executed a Commission Writeoff from the <code>Commission Writeoff</code> wizard that is accessed by clicking <code>Actions</code> on the <code>Producer Summary</code> screen and selecting <code>Writeoff Commission</code>. A new authority limit to the system named <code>writeoffchargecommission</code> was added, that checks whether you have sufficient authority to perform a commission writeoff. In the wizard, you can select multiple charges against which to do commission writeoffs, and the authority limit check will be on a charge by charge basis.</p>
CC-58679	<p>A link was added <code>InvoiceItem</code> and the transactions <code>CmsnTransferredTo</code>, <code>CmsnTransferredFrom</code>, and <code>ReserveCmsnEarned</code> subtypes. This link was already on the producer transaction, but was not being set. This does not appear in the BillingCenter base application, but the call <code>txn.getInvoiceItem()</code> will return the correct result.</p>
CC-58760	<p>A Null Pointer Exception used to occur when trying to writeoff a Commission where some of the Invoice Items for a Charge have been transferred to another producer. BillingCenter now filters out nulls from the <code>ItemCommissions</code> List.</p>
Core - Platform	
PL-9041	Modified how the currency amount data type works. You no longer specify <code>secondaryAmountProperty</code> and <code>exchangeRateProperty</code> on columns of type <code>currencyamount</code> . Instead, use the new <code>CurrencyAmountComponent</code> if you want a quantity that is editable in the application interface.
PL-10533	Fixed an issue with the regen-pcfmapping tool that caused it to not generate some display labels.
PL-10601	Corrected an issue that caused the <code>regen-toolkit</code> command to fail due to an issue with how the application handled third-party JAR files
PL-10749	Provided the ability for extensions to support linguistic search on a base column by providing a <code>supportsLinguisticSearch</code> attribute on the <code><column-override></code> element.
PL-10761	Corrected a GosuDoc issue that generated a <code>NoClassDefFoundError</code> exception. Even if a <code>gosudoc.properties</code> file existed in the configuration module, the application did not read it.

Issue ID	Description
PL-10874	Modified how the application handles caching of zone data. Previously, large amounts of data would create a severe performance bottleneck and had the potential to crash the application server. This performance fix greatly mitigates these issues.
PL-10896	Disabled the running of validation rules for importing SystemTables.xml and bootstrap.xml. Guidewire still runs validation rules on files in config/import/gen, which are modifiable.
PL-11001	Corrected an issue in which the gw.api.util.Logger utility did not consistently log the server name to the console and in the log file.
PL-11015	Guidewire has created a new server mode, <i>test</i> , that behaves exactly like a production server, except for the ability to modify the system time through an implementation of ITestingClock.
PL-11053	Dates with trailing characters were previously allowed and parsed with potentially the wrong year. Date strings with trailing characters are now rejected.
PL-11062	Moved gw.plugin.util.RemotableSearchResultSpec and gw.plugin.util.RemotableSortColumn out of util. This change means that they are no longer accessible in Java. Instead, for Java, use com.guidewire.bc.plugin.util.RemotableSearchResultSpec and com.guidewire.bc.plugin.util.RemotableSearchColumn.
PL-11077	Corrected an issue that caused a Runtime Exception if you did the following sequence of events: <ul style="list-style-type: none"> Created a user validation rule that references a custom system permission. Dropped the application database. Started the application, which triggered a database upgrade.
PL-11078	Guidewire officially deprecates library package gw.api.domain.ServiceInfoSource. In its place, use package library gw.api.system.server.ServerUtil.
PL-11136	Corrected an issue with the configuration upgrader that caused it to throw an exception and quit. This happened specifically if you were running an upgrade within a major release to a maintenance release.
PL-11162	Corrected an issue that caused the application to fail to resize an encrypted column properly if you changed the encryption plugin.
PL-11278	Modified the Data Dictionary so that Yes/No fields reflect the actual column type, which is bit (not Boolean).
PL-11283	Modified the Data Dictionary so that the display name for a foreign key is the columnName attribute for that column.
PL-11284	Changed the Data Dictionary to display the columnName attribute (not the name attribute), if a <column> element contained both. The dictionary now displays the name attribute if both exist.
PL-11285	Removed the set of links that existed underneath the Subtypes lists (if a data entity had subtypes). The links referenced Fields, Arrays, Subtypes, and Unique Keys. The list of Unique Keys was not unique.
PL-11286	Corrected an issue that caused the Data Dictionary to display derived fields and arrays in inappropriate sections.
PL-11312	Corrected an issue that caused the Data Dictionary to reference custom entity database tables incorrectly.
PL-11549	Corrected an issue with the (Server Tools) Guidewire Profiler so that it now correctly displays the proper number of milliseconds. Previously, there were cases in which this value was incorrect.
PL-11634	Corrected an issue in which the application ignored the reject methods on validatable extension entities.
PL-11709	Added a histogram to the column EditEffectiveDate in EffDatedBranch.eti.
PL-11739	Introduced a way to suppress conversion of containers (arrays, lists, maps) when calling Java code. This only is available if the classes do not use Guidewire entity instances (or containers that contain entity instances). To use this feature, underneath the BillingCenter/modules/configuration/plugins/shared directory, create a new directory called <i>basic</i> . Put your libraries and classes in subdirectories <i>classes</i> and <i>lib</i> , just like other plugins directories.
PL-11819	Corrected an issue with the Studio debugger that caused an exception to occur during the logging of the SQL statements made by Gosu find expressions.
PL-11835	Modified the data model to allow extending a non-extendable entity provided that the only changes are to validator or logicalSize parameters using the <column-override> element. This usage replace the <FieldValidatorOverride> and <FieldLengthOverride> elements, which have been removed from fieldvalidators.xml.
PL-12257	Guidewire has modified how the application handles caching of zone data. Previously, large amounts of data would create a severe performance bottleneck and had the potential to crash the application server. This performance fix greatly mitigates these issues.

Issue ID	Description
Delinquency Plan	
CC-58716	Corrected an issue where the delinquency process assign by round robin was causing an error.
Document Management Subsystem	
CC-57679	Fixed an issue where if a custom plugin is used for Search RemoteableSortColumn, a Java exception was raised when you clicked the document search results to sort based on that column.
Documentation	
CC-58228	The description of suspense payment allocation in the <i>BillingCenter Application Guide</i> has been modified as follows: Suspense payment amount is first applied to the account. The money will be distributed to charges according to BillingCenter distribution logic. The suspense payment is not reserved for the specific policy.
CC-58654	The billing plan setting Try to pay delinquent amount with funds from sibling policies has been removed.
Equity Dating	
CC-58530	Previously, the getPaidThroughDate() method returned null when the current date was before the policy's effective date. The clause that caused this behavior has been removed.
General integration	
CC-58583	Fixed an issue when Calling BillingCenter web services from PolicyCenter when the web service has a breakpoint in Studio caused an exception.
Gosu	
PL-10778, PL-12585	This change fixes a bug that was introduced in Gosu in BillingCenter 2.1.6 that caused the arguments of a method call to be evaluated before the object target expression of the method call. In nearly all cases this causes no change in behavior. In extremely rare cases, if there were side effects of the argument expressions that change the object target expression value, the behavior is now consistent with the original semantics.
PL-11044	Added a new server configuration option, called RetainDebugInfo, to facilitate debugging from Studio without a type system refresh. If set to true, GosuClasses does not clear debug information after compilation and the server is always considered to be debuggable.
PL-11845	Corrected a Studio debugger issue that caused the debugger to not stop on a breakpoint if running in an integrated BillingCenter/PolicyCenter environment.
Integration	
PL-10503	The update Archive batch process name and description have been changed to reflect the new archiving framework.
PL-10599	Beginning with the release of BillingCenter 3.0.0, you might experience namespace collisions in the names of argument types and return types on published web services. If this occurs, Studio notifies you with errors. To make debugging during upgrade easier, BillingCenter 3.0.2 has a new configuration parameter called AllowSoapWebServiceReferenceNamespaceCollisions. If set to true, these error messages become warnings. Use this for development and debugging until you have time to rename your classes to fix the namespace collision. This setting is false by default. Guidewire does not support setting this value to true for production servers. (It is unsafe to do so.) You can contact Guidewire Customer Support for details.
PL-10629	Guidewire has modified the way it supports InetSoft VPM. This work has tightened the Guidewire SOAP security for ISREAuthenticationAPI and added three more properties to sree.properties. <ul style="list-style-type: none"> • gw.soapuser.username: The user name used by InetSoft to retrieve user/group/role information used by VPM and authentication • gw.soapuser.userpassword: The password associated with the user name. • gw.usercache.size: The cache size for user cache, group cache, and role cache.
PL-10972	Application server now logs an error whenever its connection status to the InetSoft report server changes.
PL-11193	Added a UseMessageCreatorAsUser property to the emailMessageTransport plugin. By default, the emailMessageTransport plugin will use system user to retrieve a document. Set this value to true to retrieve the document on behalf of the user who generated the email message.
PL-11221	Corrected an issue with the IMessagingToolsAPI API that occurred while attempting to acknowledge messages using web services. This issue occurred if the number of field changes was bigger than the smallest number of field values in each field change.
PL-11375	Corrected a security issue with InetSoft reports. The fix included using username credentials instead of client information for principal creation.

Issue ID	Description
Manageability - Platform	
PL-10035	Modified the (Server Tools) Data Distribution page and added several new non-Boolean types to track the size of a group of values and the number of groups with that size.
PL-10984	Modified the (Server Tools) Info Pages → Data Distribution screen to ensure that the user selects at least two data distributions to include in the download.
PL-11038	Corrected a caching issue with the Sequence Generator that caused it to work incorrectly.
PL-11127	Modified the (Server Tools) Data Distribution page to disable the Download Zip File button if no background data distribution batch process is running.
PL-11153	Added a new download to the (System Tools) Work Queue page. This download provides a zipped CSV file with the history of a given work queue (timing, completed items count, and throughput).
PL-11244	Added a new (Server Tools) Database Statistics screen that you can use to view previous database statistics based on a given date.
PL-11427	Added a new <code>sqlserverCreateIndexSortInTempDB</code> attribute to the <code><upgrade></code> element of the <code><database></code> element in the <code>config.xml</code> file. The default is <code>false</code> . If true, the attribute enables the ability to use tempdb for sorting if building indexes in SQL Server. Refer to http://msdn.microsoft.com/en-us/library/ms188281.aspx for details on the requirements to use tempdb for sort results.
PL-11431	Converted all backing indexes for all primary keys to clustered indexes in SQL Server. The change improves performance of archiving and purge operations. During a database upgrade, the upgrader automatically converts the backing indexes for all primary keys to clustered indexes and drops any clustered indexes on tables with a primary key.
Notes	
CC-57799	Policy and producer notes are now displayed in row format.
Payment	
CC-53364	You can now override the criteria by which payment requests are sent out. A new method was added to the Invoice plugin: <code>boolean shouldCreatePaymentRequest(AccountInvoice invoice)</code> . The default is for a payment request to be sent when the payment method is non-responsive.
CC-55611	The following API changes were made related to money received handling: <ul style="list-style-type: none"> Added a <code>PolicyPeriod</code> field to the <code>DirectBillMoneyRcvd</code> entity which is a reference to the policy period to target during the initial distribution. For the <code>IDirectBillPayment</code> plugin interface and the built-in implementation of it, the method signature for <code>allocatePolicyPeriod(...)</code> changed. It now has an additional parameter for a set of other eligible invoice items. These are the other invoice items that are eligible to be paid that are not associated with the policy period. All items that are eligible to be paid and associated with the specified policy period are already on the payment and can be retrieved by calling <code>getDistItems()</code>. The <code>DirectBillPaymentFactory</code> class now has two new methods: <code>createDirectBillMoneyRcvdWithPolicyPeriod(Account, PaymentMethod, Amount, PolicyPeriod)</code> and <code>createAndExecutePaymentTargetingPolicyPeriod(DirectBillMoneyRcvd)</code>
CC-57202	In the New Payment screen for direct bill payments, the payment description no longer disappears when you modify a payment.
CC-57697	The Account → Payments screen for direct bill payments now has a reversal date field.
CC-58038	Fixed an issue on the Account → Payments screen, where the Distributed field was set to Yes for suspense items even when there was no distribution.
CC-58110	It is now possible to select a custom group criterion for grouping invoice items on the New Payment screen for direct bill payments. To support this functionality, the <code>CustomPaymentGroup</code> field was added to <code>InvoiceItem</code> . Also, the <code>Custom Group</code> typekey was added to <code>DBPmntGroupingCriteria</code> typelist.
CC-58152	Previously, ascending and descending queries for <code>ReceivedPaymentMoneys</code> returned the same results. The <code>findReceivedPaymentMoneysSortedByReceivedDate</code> and <code>findReceivedPaymentMoneysSortedByReceivedDateDescending</code> methods were fixed to implement the expected sorting. This also fixed the sorting on the New Payment screen for direct bill payments.
CC-58539	The <code>OriginalPayment</code> is a foreign key to <code>PaymentMoneyReceived</code> (virtual property). Previously, this foreign key was not being set when a payment was moved to a different account. This has now been fixed.
CC-58541	When BillingCenter receives a renewal billing instruction, the associated suspense payment is applied to the new policy period.

Issue ID	Description
CC-58559	Extension fields (other than links or arrays) are now preserved on payment items during modification of payment and promise items.
CC-58608	This fix ensures that no distribution is created if no payment items are allocated. If the implementation of the DirectBillPayment plugin has been configured such that a DirectBillPayment handed does not allocate funds against any items or suspense items, the DirectBillPayment will be retired and will not mark its associated DirectBillMoneyRcvd as having been distributed.
CC-58609	In the BillingCenter base application, you can modify a direct bill credit distribution from the user interface.
CC-58629	Previously, the Automatic Disbursement batch process created an additional record for a Direct Bill Payment every time it ran. This behavior has been changed so that a new modifying payment is created only if the payment items or money received has changed in some way (for example, items added or removed, existing items changed, or the money received fields changed). If the modifying payment does not differ from the original, it is removed and the original payment continues to be the payment of record.
CC-58638	Extension fields (other than links or arrays) that have the same name in both BaseMoneyReceived.etc and SuspensePayment.etc are copied to a suspense payment. When the suspense payment is converted to an actual payment, the extension fields are copied to the payment.
CC-58830	The add.gif image, which used to indicate items that were manually added in the Direct Bill Payment wizard, was accidentally removed in 3.0.0. This has now been restored.
Payment Distribution	
CC-58557	The Invoice batch process was distributing payments that are reversed and undistributed. This has been corrected.
Performance	
CC-58558	The default behavior of generating transaction numbers has changed. To improve performance, transaction numbers may no longer be sequential second to second, but will be mostly sequential hour by hour. Previous transaction numbers were 10-digit numeric strings. Now, transaction numbers are 10 digits followed by a dash, followed by 4 digits. Details of the implementation can be seen by reviewing SequenceNumberGenerator.gs. Instructions on how to revert to the old behavior can be found in the comments of that file.
Policy File	
CC-58666	Previously, equity dating calculations, including both getPaidThroughDate and getPolicyEquity, were not taking policy-level writeoffs into consideration. A new configuration parameter: EquityDatingIncludeWriteoffs has been added. The default value for this parameter is false. <ul style="list-style-type: none"> • If set to true, the amount of policy writeoffs is added to the payments when calculating equity and paid through date. • If set to false, the calculations do not include writeoffs.
Producer	
CC-58532	The first screen of the Commission Writeoff Recovery wizard no longer throws an exception.
Reporting	
CC-57334	The additional extra commission reserve transaction is no longer created on a Premium Report charge.
CC-58351	This fixes a report permission problem with the BillingCenter / InetSoft integration that could, under somewhat rare circumstances, prevent users from running reports in BillingCenter.
CC-58467	The Account/Producer Balance report now displays 0 instead of a blank to render a null value.
CC-58555	The calculation of Beginning Outstanding Amount has been fixed in the Account Reconciliation report. The paid amount (instead of the transaction amount) is now being compared against the invoice amount.
CC-58664	In the Account Balance report, the Other Credits column in the report was incorrectly showing disbursements. This has been fixed.
CC-58678	Account Balance report now shows the negative writeoffs amount under Debts and includes NegativeWriteoffAcct transactions in its calculation.

Issue ID	Description
CC-58731	The Producer Balance report has been modified to ensure it is balanced as follows: <ul style="list-style-type: none"> • AgencyMoneyReceivedTxn now adds to the AB Payments section of report. • ChargePaidFromProducer (Unapplied line item) now properly adds to the Charges Paid section of report. • ProducerWriteoff transactions now properly update the Writeoffs and Negative Writeoffs on the report. (Positive write-offs used to show a negative amount.) • CmsnPayableReduction and CommissionWrittenOff transactions with negative amounts now properly update the Negative Writeoffs on the report. (They used to be considered Writeoffs). • CmsnRivableReduction and CommissionWrittenOff transactions with positive amounts now properly update the Writeoffs on the report. (They used to be considered Negative Writeoffs).
CC-58790	AB Charges Paid and AB Commission Paid now appear in the Debits section of the Producer Balance report.
Studio	
CC-58281	Previously, when you created a new XML model for an entity, the Amount field did not appear in the GX Tools modeler list of properties. This happened because the getAmount method was not specified as scriptable. The method is now scriptable and appears in the list.
PL-10380	Modified the behavior of the Messaging editor so that the Remove button is unavailable if there is only a single defined messaging environment.
PL-10442	Corrected an issue that prevented Studio from moving a PCF file between folders if its name was not also changed.
PL-10447	Corrected an issue that caused a ClassNotFoundException (on AbstractFireTest) upon trying to start the GUnit Debug utility.
PL-10455	Corrected an issue in which using CTRL+left/right arrow to move the cursor from one word to another treated characters accents, cedillas, umlauts, and other modifiers as word breaks (in the same way it treats spaces). This happened with é, ç, and è, for example.
PL-10530	Added the ability to manage certain CSV (comma separated value) files within Studio. (These files are in the Other Resources folder and include authority-limits.csv, activity-patterns.csv, reportgroups.csv, and similar files.) Studio manages the file (adding an editable copy to the configuration folder, for example). You must however, edit these files using an external editor.
PL-10570	Corrected an issue that caused an Illegal ordering found for element InputFooterSection error. This occurred if you added an InputColumn widget to the far right of an existing InputColumn widget in a PCF containing an InputFooterSection and tried to save the change.
PL-10598	Modified the behavior of the PCF TypeKeyInput/RangeInput widget's filter so that it now recognizes VALUES as a valid value.
PL-10605	Modified the behavior of the Studio Entity Names editor so that it now properly displays entities with displayName columns, and makes the symbols used in the Entity Name Gosu editor into String types for displayName="true" columns.
PL-10698	Modified the behavior of the Create Display Key dialog so that it now wraps text automatically as needed.
PL-10728	Corrected an issue that caused a ClassNotFoundException error if you attempted to revert a newly created PCF file within source control using the Studio File → Source Control → Revert menu.
PL-10734	Modified the Rule Conditions editor to display an error if the condition is not a boolean expression. (For example, an assignment statement such as account.AccountType = "payer" is not a boolean expression and should display an error.)
PL-10806	Modified the behavior of the Go-To (CTRL-N) pop-up dialog so that you can now search on workflow names, entity names, and plugin names.
PL-10835	Corrected an issue in which Studio did not display a warning on the use of improper case in Gosu templates. This occurred, for example, if you defined a variable with an initial upper-case character and then you later used that variable with all lower-case characters.
PL-10919	Modified the Messaging editor to disable the Add button if the Env field is empty (meaning no name exists in the Env field).
PL-11023	Corrected an issue with typing text in the New Workflow Subtype Metadata dialog.
PL-11024	Corrected an issue that permitted recursive screen references on a PCF page, causing a stack overflow.
PL-11137	Modified the extension file that Studio automatically generates when you select an entity .eti file, right-click, and select the Create extension file option. Studio now creates separate <extension> and </extension> tags, making the file easier to modify.

Issue ID	Description
PL-11195	Corrected an issue that caused Studio to truncate the display key text used to generate widget titles in PCF files.
PL-11225	Updated Studio so that the (File) Import and Export submenus are correctly localized, if you install a localization module.
PL-11339	Corrected a timing issue that required you click OK twice to close the Display Key Editor dialog that opened from the PCF editor.
PL-11345	Corrected an issue in which the GUnit tester failed (and generated an error) if you attempted to reinvoke a test from the Result window.
Suspense Payment	
CC-57834	The Payment Date for suspense payments on the Search Payments screen is now hyper-linked to the Suspense Payments screen on the Desktop.
CC-57836	Check # field added to the Desktop → Suspense Payments screen and to the Edit Suspense Payment screen.
Tools	
CC-58545	GX Tools can now populate a one-to-one field.
TroubleTicket	
CC-57797	The Account, Policy, and Producer entities are now hyper-linked on the Trouble Ticket screen.
Web	
PL-10438	Corrected an exception if you revisited an uncommon search page whose search result is not Query Result type.
PL-10476	Added the ability to generate a new row in ListView edit mode by selecting ENTER in the last editable cell.
PL-10706	Corrected an issue that generated a NullPointerException if you called an addFieldError method from a validate method invoked from certain PCF pages.
PL-11227	Corrected an issue that the application did not correctly apply the useArchivedStyle property to a TextAreaCell1 widget. (The cell still shows the standard font color.)
Write-Off	
CC-58579	In the New Writeoff wizard, you can change your selected policy period as the target of the writeoff without causing an exception.

Known Issues and Limitations

Note: For maintenance releases, it is Guidewire policy to defer fixing configuration issues if they require merging files during the upgrade. Workarounds to many of these issues are listed in the following sections. The goal of this policy is to make upgrades as straightforward as possible.

This section describes known issues with this release of Guidewire BillingCenter.

Acrobat Sample document template does not allow extra fields (PL-1416)

Issue: Currently, the example Acrobat document template uses two fields, and this requires the same number of fields in `SampleAcrobat.pdf.descriptor` and `SampleAcrobat.pdf` files.

Workaround: To allow more fields in the template and documents created from it, you must update its descriptor file.

Server exception during startup (PL-8167)

Issue: During server startup, there can be a Tomcat server exception. However, after the exception, the server does start up. This is caused by the Tomcat application server attempting to either save or restore sessions using serialization. The session restore causes an exception because BillingCenter sessions contain objects which rely on the metadata being started up.

Workaround: None. The Tomcat application server does start up after the error.

Zone data with missing values cannot be loaded (PL-9123)

Issue: Zone data, which is used for associating postal codes with cities and regions, does not load if it is incomplete.

Status: Guidewire is aware of this issue.

Not Possible to Hot-Deploy on all Supported Application Servers (PL-9955)

Issue: It is not possible to hot-deploy on all supported application servers.

Workaround: If you use Tomcat, you are able to see the changes without requiring a restart of the application server. If you use an other application server, you need to perform a restart to see any configuration changes reflected.

Values in MIME-Type Drop-Down Are Not Localized (PL-10921)

Issue: BillingCenter does not localize the values in the `MIME type` drop-down list properly.

Workaround: If you want to localize MIME types, you can do so by modifying the definition in `config.xml`. Currently, BillingCenter does not support MIME type definitions in multiple languages.

Finder in Annotation Causes Throws Error (PL-10812)

Issue: Studio throws error and fails to do syntax error highlight, code completion, and similar methods, if there is a finder in an annotation. This occurs even if you are connected to the server.

Workaround: Remove the finder from the annotation.

Navigating Between Pages in Wizard Loses Data (PL-10920)

Issue: In a wizard pop-ups, it is possible to lose data if you navigate to another page in the wizard.

Workaround: Navigate out of the wizard and then back into it rather than changing steps directly in the wizard.

Studio Does Not Properly Report Verification Error (PL-10822)

Issue: Studio does not properly report a verification error if a typecode contains a trailing space.

Workaround: Remove trailing spaces from all type codes.

Double Quotes in Path Environment Variable Causes Problems (PL-10981)

Issue: BillingCenter does not run if there are double quotes in the `PATH` environment variable.

Workaround: Check the `PATH` variable and remove any double quotes.

Studio Debugger Does Not Handle Rule Breakpoints Properly (PL-10553)

Issue: If you set a breakpoint in a Gosu rule and attempt to step through the code using the Studio debugger, the debugger does not correctly display the code on the breakpoint step.

Status: Guidewire is aware of this issue.

Inability to Access Typecodes that Contain a Dash (PL-10532)

Issue: It is not possible to access a typekey code (typecode) that contains a dash using Gosu.

Workaround: Remove any dashes from the typekey codes.

Linguistic Search Performance on Oracle is Slow (PL-12143)

Issue: BillingCenter uses an Oracle Java Stored Function to perform linguistic searches. Such search queries can perform poorly. BillingCenter invokes the stored function on Oracle if either one of the following is true:

- The primary locale language is German or Japanese
- The `LinguisticSearchCollation` strength is set to `primary` in the default locale's `GWLocale` element in file `localization.xml`.

This is an issue for any linguistic search in which you set the `LinguisticSearchCollation` strength to `primary`. For example, if you are working in a language that commonly uses accents (such as French or Spanish), and you wish to perform accent-insensitive searches, then you typically set the `LinguisticSearchCollation` strength to `primary` for that language. If you do so, it is possible to experience slow performance on an Oracle database as you perform linguistic searches.

Workaround: Guidewire recommends that you set the `LinguisticSearchCollation` strength to `secondary` if applicable. (This is not applicable for Japanese and German.)

InetSoft Exception Failed Login Exception: Bad user name anonymous (PL-10148)

Issue: InetSoft generates a `Failed Login Exception: Bad user name anonymous` exception on server start-up. This occurs if you set the InetSoft logging level to `fine`.

Workaround: This error does not cause any issues. You can safely ignore the exception.

Report Still Visible After Permissions Disabled For Parent Folder (PL-10353)

Issue: An InetSoft report that is included in a permission set is still visible in the report tree even after its parent has been removed from the permission set.

Workaround: You must remove individual reports from permission sets instead of maintaining the reports at the report folder level.

ToolbarButton Widget Does Not Render Correctly (PL-10742)

Issue: BillingCenter incorrectly renders a `ToolbarButton` widget if the associated `AddMenuItemIterator` is empty.

Workaround: Modify your configuration to disable the menu button if the button contains no menu item.

Must Name Row Iterator Element with Capital Letter (PL-10795)

Issue: The Document search results page (`DocumentLV.pcf`) throws a `NullPointerException` if the row iterator element name is `document` (with a lower case d).

Workaround: You must use `Document` (with a capital D) instead.

Large Number of Classes in plugins\shared\lib Causes Studio to Fail to Start (PL-11026)

Issue: If you attempt to load multiple JAR files that contain many thousands of classes, Studio fails to start and throws an `OutOfMemory` error.

Workaround: Increase `MaxPermSize` to something over `128m` (the default) so that Studio has enough memory to build the type information database.

ANT Command `build-war` Hangs Indefinitely (PL-11045)

Issue: It is possible for the `build-war` ANT command to hang indefinitely. This issue is only reproducible on certain environments.

Workaround: Set the `-noinput` parameter before calling the `build-war` ANT command. For example, instead of `gwcc.bat build-war`, do the following:

```
gwbc.bat -noinput build-war
```

Missing Display Keys Cause Server to Fail to Start (PL-11764)

Issue: Missing display keys defined in the application default locale can cause the server to fail to start.

Workaround: Within Studio, run the **Verify Resources** utility before attempting to start the application server. BillingCenter bases the type system for display keys on those defined in the default application locale. If a display key in the default application locale is missing and a rule references the missing display key, then BillingCenter cannot compile the rule. Running the resource verification utility can catch these types of missing display keys.

BillingCenter Not Generating `soap.local.entity` Correctly (PL-7560)

Issue: BillingCenter is not generating the `soap.local.entity` for a class used by web services.

Workaround: To ensure that BillingCenter generates the `soap.local.entity` correctly for `FieldValue` classes (which are Java types), subtype `FieldValue` into `GosuFieldValue`, then further subtype that class.

Incorrect additivity statements in the default `logging.properties` file (CC-50615)

Issue: Some additivity statements in the `logging.properties` file are incorrect and therefore, ineffective. An additivity statement attempts to keep the category from sending its log messages up to its parent, preventing duplicate log messages across files.

Workaround: In the `logging.properties` file (located at `modules/cc/config/logging`), change:

```
log4j.additivity.PluginsLog=false  
to  
log4j.additivity.Plugin=false
```

BillingCenter login remains active in new browser window (CC-4152)

Issue: When you log into BillingCenter in a browser window, then you close that window and open a new one, in some instances you can access BillingCenter pages without needing to log in again.

This is an issue with some versions of Microsoft Internet Explorer. When multiple browser windows are running in the same process, they can share session information, and this allows them to share a single login with BillingCenter. Early versions of Internet Explorer provided a **Launch browser windows in a separate process** option, which, if enabled, would avoid this situation. Some versions of Internet Explorer do not allow you to set this option, and enable or disable it automatically. You can review the relevant Microsoft Knowledge Base Article 240928. However, you may experience a situation in which Internet Explorer does not correctly enable this option, especially if you manually disabled it in an earlier version and then later upgraded. In this case, the multiple browser windows are run in the same process, exhibiting the undesired behavior.

Workaround: The simplest solution is to always click **Logout** at the top of the BillingCenter window before closing your browser.

You can also take steps to manually enable the Internet Explorer option. This requires you to edit the Windows registry on the affected computer.

WARNING Editing the registry is a risky procedure, and any improper changes may result in permanent data loss and require Microsoft Windows to be completely reinstalled on that computer. To make this change, create or edit the

`[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\BrowseNewProcess]` registry entry. Set the data type to `REG_SZ`, and then set the data value to `yes`.

[WebSphere timeout during initial startup \(CC-4904\)](#)

Issue: The initial creation of the BillingCenter database may take longer than the default WebSphere startup timeouts will allow, and WebSphere may abort the startup before the database upgrade has completed. In this situation, the `Default_Server_stderr.log` file shows an interruption in the `Applying upgrade step #### of #####` log messages.

Workaround: Increase the Ping Timeout and Ping Initial Timeout values for WebSphere. These can be found in the WSADMIN Console. Click **Nodes**, drill down to your application server, click the **Advanced** tab, and increase the values.

[BillingCenter displays many warnings when regenerating Javadoc \(CC-55564\)](#)

Issue: This issue occurs when running the `regen-java-api` and `regen-soap-api` command.

Status: Guidewire is aware of this issue.

[On AIX without an X11 server, BillingCenter may report that an X11 resource is missing \(CC-5889\)](#)

Issue: BillingCenter does not require an X11 server; however other components of the BillingCenter environment (such as Java) may inadvertently have this requirement. If so, BillingCenter may report that an X11 resource is missing when it is starting up.

Workaround: Install an X11 server, or contact Guidewire support to determine if the dependency can be removed.

[Port conflict when starting BillingCenter with JMX RMI adapter enabled and Microsoft Outlook running \(CC-6964\)](#)

Issue: When Microsoft Outlook starts, it chooses several ports to use for its connection to an Exchange server. In some instances, Outlook may use port 1099, which is a port that the BillingCenter JMX RMI adapter can use, *if enabled*. The BillingCenter server then reports a port conflict during startup when it tries to use that port for the RMI adapter.

Workaround: Close Outlook to free the port, and then start the BillingCenter server again. You can also change the port that the RMI adapter uses by editing the `JMXManagementPlugin` through Guidewire Studio.

[Studio PCF Editor treats commented-out property values as errors \(PL-4582\)](#)

Issue: If you attempt to comment out a property value in the Studio PCF editor, Studio treats this as an error.

Status: Guidewire is aware of this issue.

[Incomplete implementation of non-English locale-specific database search and sort \(PL-8108\)](#)

Issue: Guidewire has begun implementation of database searching and sorting appropriate to non-English locales. This release of BillingCenter contains an incomplete implementation of this work. The changes will not affect customers implementing case-insensitive searches using standard English sort criteria.

As part of this work, the current release renames the `supportsCaseInsensitiveSearch` attribute on column to `supportsLinguisticSearch`.

Status: Guidewire is aware of this issue.

[Creating Standard XSDs and Standard XML documentation is missing from the HTML documentation \(CC-58429\)](#)

Issue: BillingCenter exports business data entities, Gosu class data, and other types to a standard Guidewire XML format, using a feature called GXTools. Due to a processing error, the documentation for this feature is not available in the HTML version of the documentation.

Workaround: You can read about this feature in the PDF version of the documentation. See Gosu Reference Guide → Gosu and XML → Creating Standard XSDs and Standard XML.

[Server exception during startup \(PL-8167\)](#)

Issue: During server startup, there can be a Tomcat server exception. However, after the exception, the server does start up. This is caused by the Tomcat application server attempting to either save or restore sessions using serialization. The session restore causes an exception because BillingCenter sessions contain objects which rely on the metadata being started up.

Workaround: None. The Tomcat application server does start up after the error.

[Zone data with missing values cannot be loaded \(PL-9123\)](#)

Issue: Zone data, which is used for associating postal codes with cities and regions, does not load if it is incomplete.

Status: Guidewire is aware of this issue.

[H2 development database creates LONGTEXT fields as VARCHAR\(65000\), instead of as CLOB as in Oracle and SQL Server \(PL-9314\)](#)

Issue: Guidewire creates schema fields defined as LONGTEXT as LONGVARCHAR columns, instead as CLOB as in the Oracle or SQL Server databases. In addition to the inconsistency involved, this mandates a field length of 65,000 characters or less in LONGTEXT columns in the H2 database.

Workaround: Define a LONGTEXT field as a VARCHAR(...) with the required size, for example, as VARCHAR(120000).

[Report Folders Do Not Display If Empty \(PL-1281\)](#)

Issue: Guidewire BillingCenter does not display a report folder if that folder is empty.

Workaround: Place an InetSoft replet inside the folder. You can mark the replet as *not visible* within the InetSoft Enterprise Manager. This ensures that the parent folder displays in BillingCenter even if the folder is empty.

[Changes to encryption column can cause encrypted columns to shrink \(PL-9665\)](#)

Issue: If you change the encryption algorithm in such a way that the encrypted columns shrink, then it causes the resulting upgrade to crash on SQL Server.

Status: Guidewire is aware of this issue.

[Report Folders Do Not Display If Empty \(PL-1281\)](#)

Issue: BillingCenter does not display a report folder if that folder is empty.

Workaround: Place an InetSoft replet inside the folder. You can mark the replet as *not visible* within the InetSoft Enterprise Manager. This ensures that the parent folder displays in BillingCenter even if the folder is empty.

[Not Possible to Hot-Deploy on all Supported Application Servers \(PL-9955\)](#)

Issue: It is not possible to hot-deploy on all supported application servers.

Workaround: If you use Tomcat, you are able to see the changes without requiring a restart of the application server. If you use an other application server, you need to perform a restart to see any configuration changes reflected.

Values in MIME-Type Drop-Down Are Not Localized (PL-10921)

Issue: BillingCenter does not localize the values in the **MIME type** drop-down list properly.

Workaround: If you wish to localize MIME types, you can do so by modifying the definition in `config.xml`. Currently, BillingCenter does not support MIME type definitions in multiple languages.

Finder in Annotation Causes Throws Error (PL-10812)

Issue: Studio throws error and fails to do syntax error highlight, code completion, and similar functions, if there is a finder in an annotation. This occurs even if you are connected to the server.

Workaround: Remove the finder from the annotation.

Navigating Between Pages in Wizard Loses Data (PL-10920)

Issue: In a wizard pop-ups, it is possible to lose data if you navigate to another page in the wizard.

Workaround: Navigate out of the wizard and then back into it rather than changing steps directly in the wizard.

Studio Does Not Properly Report Verification Error (PL-10822)

Issue: Studio does not properly report a verification error if a typecode contains a trailing space.

Workaround: Remove trailing spaces from all type codes.

Double Quotes in Path Environment Variable Causes Problems (PL-10981)

Issue: Guidewire BillingCenter does not run if there are double quotes in the **PATH** environment variable.

Workaround: Check the **PATH** variable and remove any double quotes.

Studio Debugger Does Not Handle Rule Breakpoints Properly (PL-10553)

Issue: If you set a breakpoint in a Gosu rule and attempt to step through the code using the Studio debugger, the debugger does not correctly display the code on the breakpoint step.

Status: Guidewire is aware of this issue.

Inability to Access Typecodes that Contain a Dash (PL-10532)

Issue: It is not possible to access a typekey code (typecode) that contains a dash using Gosu.

Workaround: Remove any dashes from the typekey codes.

Incorrect totals in the policy payment schedule occurred on agency bill statements after a change was made to one of two producers on a policy (CC-56925)

Issue: The producer statement does not show the correct dollar amount after a change is made to one of the policy's payment plans on a policy with two producers after the statement was billed and partially paid.

Status: Guidewire is aware of this issue.

BillingCenter did not process policy changes done after the policy expiration date (CC-56852)

Issue: During BillingCenter - Policy Center integration, BillingCenter does not process policy changes made after the policy expiration date. An illegal state exception displays on both the BillingCenter and PolicyCenter consoles.

Status: Guidewire is aware of this issue.

Moving a Billed invoice item should negate the direct bill commission paid, not the agency bill commission paid (CC-56656)

Issue: If an item is moved from a Billed invoice to a Planned invoice, an earnings transaction is created for the Payable amount on the commission. However, if the commission was earned and paid, it is not being negated.

Status: Guidewire is aware of this issue.

After a reversal, the distributed payment is still being reserved for the premium payment (CC-56538)

Issue: The disbursement amount is not fully moved to disbursement and does not show up on the Disbursement page although an entire payment has been reversed and the associated charge had been assigned to another account.

Workaround: This is a new feature that is not fully implemented. Guidewire is aware of this issue.

The applied net amount on an agency bill statement was not accurately calculated after reassignment of a partially paid item (CC-56153)

Issue: The payment applied net amount on an agency bill statement was not correctly calculated after a partially paid agency bill item was reassigned to another producer.

Workaround: Guidewire is aware of this issue.

Errors occur in the totals for the producer's rate on the Transactions screen (CC-55835)

Issue: A difference occurs in the totals shown on the Transactions screen for a producer when selecting Reserves and then choosing either the Last 30 Days or Show All filter.

Workaround: Guidewire is aware of this issue.

New collateral charges and invoice amounts are being generated based on previously paid collateral amounts, not the previously billed or planned amounts (CC-55209)

Issue: New collateral charges and invoice amounts are being generated based on the amount of collateral previously paid and do not account for amounts of collateral previously Billed or Planned (not yet billed). This results in an over billing of the collateral amount on the invoices.

Workaround: Guidewire is aware of this issue.

During the charge invoicing process, the producer statements for the month of February were not being created (CC-54952)

Issue: When creating a new agency bill policy with a monthly payment plan, ten installments, and a down payment the February statement was not generated. Instead, the February item was placed on the invoice for the next month.

Workaround: This is part of a new feature that is not fully implemented. Guidewire is aware of this issue.

No commission reserve and onset transfer transaction was created when a direct bill item was reassigned to an agency bill producer (CC-54923)

Issue: When a direct bill policy has no specified primary producer and one of the policy's item is reassigned to an agency bill producer, no commission reserve and onset transfer transaction created. This was because there is no initial reserve.

Workaround: Guidewire is aware of this issue.

Billing instructions for Issuance and Policy Change must work with premium and other charges (Premium Reporting Only) (CC-52032)

Issue: This issue is for Premium Reporting policy periods only. Users are note allowed to enter premium charges as part of an issuance or policy change and process them as any other policy period. This issue applies to all billing instructions types.

Workaround: Guidewire is aware of this issue.

Automatic distribution to policies not working correctly (CC-51511)

Issue: Automatic distribution to policies on Account → Disbursement screen does not distribute the payment correctly.

Workaround: Guidewire is aware of this issue.

Problems occurred with the use of negative collateral charges (CC-51312)

Issue: When issuing a negative collateral charge, the amount gets subtracted from a Charge Due T-account and a payer's Unapplied T-account. This causes an imbalance between collateral reserve and collateral held when issuing a negative charge.

Workaround: Guidewire is aware of this issue.

Cannot undistribute direct bill payment (CC-58764, CC-58764)

Issue: The base configuration does not allow a direct bill payment to be undistributed.

Workaround: Use the configuration to create an Unapply button on the Payment List View. This button should remove the original DirectBillMoneyRcvd, reverse all associated transactions, and then create a brand new DBMoneyRcvd with the same attributes as the original.

Error when modifying agency bill payment (CC-58697, CC-58696)

Issue: When you try to edit an agency bill payment that is already fully paid and set the disposition to Writeoff, the following message is seen: EvaluationException: Cannot write off an Invoice Item that is already fully and exactly paid. (click for error details).

Workaround: A temporary workaround for this issue is to add reflection to disposition so that Writeoff is not available if NetAmountDue and AppliedAmount are the same.

No API to remove invoice items from DirectBillPaymentView (CC-58397)

Issue: There is no way to programmatically remove an Invoice item (or invoice item array) from a distribution.

Status: Guidewire is aware of this issue.

Invoice PreUpdate rule is bypassed when a new invoice created for Collateral charge (CC-58254)

Issue: Automatic distribution to policies on Account → Disbursement screen does not distribute the payment correctly.

Status: Guidewire is aware of this issue.

Reversing Write-Off Commissions and Write-Off Commission Recovery in the UI (CC-58655)

Issue: The UI does not provide a way reverse Write-Off Commissions and Write-Off Commission Recovery.

Workaround: Clone the Reverse button under the Payments tab and reverse CmsnPayableReduction and other transactions using the following action: CurrentLocation.startEditing(); producerTransaction.reverse(); and CurrentLocation.commit().

[Use of StringBuilder Class prevents debugging of web service \(CC-58533\)](#)

Issue: In a web service, if you have a method with a parameter of type X and inside the web service method you reference a Gosu class of type Y. And, Gosu class Y has a `toString()` method (overriding the default) that uses a `java.lang.StringBuilder` class. If you set a breakpoint in the web service method and try to step through the method, then you will get errors.

Status: Guidewire is aware of this issue.

[Should not be able to edit Payment Plan after it has been used by a policy \(CC-57758\)](#)

Issue: In the base configuration, it is possible to edit a Payment Plan even after it has been used by a policy. Only the expiration date should be updateable.

Workaround: Configure the payment plan screen to only allow the expiration date to be updated.

[Account Unapplied goes below zero when distributing moneys that have had some portion of their funds depleted \(CC-58782\)](#)

Issue: Undistributed MoneyReceived or partially distributed Payment results in Account Unapplied. If the funds are taken out of the Account unapplied, (with Account Funds Transfer, a Credit Distribution, a Disbursement, etc) and the original payment is then distributed, the Account Unapplied can dip below zero.

Status: Guidewire is aware of this issue.

[Create New Invoice button on Account Invoices Screen doesn't work \(CC-58078\)](#)

Issue: The Create new invoice button has been disabled until it can be implemented to take into account invoice streams.

Status: Guidewire is aware of this issue.

[Account Evaluation is not counting account level delinquencies \(CC-55938\)](#)

Issue: The count for number of delinquencies on Account\Evaluation screen doesn't factor account level delinquencies.

Workaround: Use the Gosu finder to identify the account level delinquency processes and add it to the BillingCenter value.

[Assertion Error when adding account charges \(of type incorrect charge pattern\) \(CC-58717\)](#)

Issue: When an incorrect Charge Pattern is used for charges (for example, Account owned pattern for Policy charges), BillingCenter allows the charges to be created, unless Java Assertion is enabled.

Status: Guidewire is aware of this issue.

[Duplicate payment plan database constraint on starting an integrated environment and sending cancelPolicy message \(CC-56972\)](#)

Issue: BillingCenter does not process policy changes done after the policy expiration date. Illegal state exception is received in console of both BillingCenter and PolicyCenter.

Status: Guidewire is aware of this issue.

BillingCenter does not process policy changes done after the policy expiration date (CC-56852)

Issue: BillingCenter does not process BillingCenter policy changes done after the policy expiration date. Illegal state exception is received in console of both BC and PolicyCenter.

Status: Guidewire is aware of this issue.

Amount Distributed on a payment does not reflect transfers.(CC-58782)

Issue: The Amount Distributed on a payment does not reflect transfers. This means that the Account : Unapplied is adjusted, but the full payment amount is still available for distribution on the original account.

Workaround: Make the following amendments to the `allocatePayment` method of the `DirectBillPayment.gs` plugin. Add the following lines to adjust the amount to allocate before you call out to the allocation strategy:

```
allocatePayment( DirectBillPayment, Amount)
if ( payment.BaseMoneyReceived.Executed) {
    amount = BigDecimalUtil.min(payment.Account.RawUnappliedAmount, amount)
} else {
    amount = BigDecimalUtil.min(
        payment.Account.RawUnappliedAmount.add(payment.BaseMoneyReceived.Amount),
        amount)
}
```

The same issue will happen when calling the following plugin methods:

- `allocateCollateral`
- `allocatePolicyPeriod`
- `allocateCredits`.

You can add the same lines of code to these plugin methods to correct the problem that occurs if the payment you are allocating has more money in it than what is actually executed against Account : Unapplied.

Some Oracle database statistics are inaccurate (CC-58861)

Issue: For Oracle databases the `bc_StandardWorkqueue` and `bc_<workItemType>WorkItem` tables are highly volatile. Frequently the database statistics on these tables indicate that there are zero rows. The optimizer can select inappropriate execution plans for some SQL statements on these tables.

Workaround: To help the optimizer make the best choice, Guidewire recommends that you delete statistics on these tables using Oracle's `dbms_stats.delete_table_stats` procedure after running the database statistics process on the application.



chapter 35

Guidewire BillingCenter 3.0.1 Release Notes

Release 3.0.1.34

Copyright © 2001-2010 Guidewire Software, Inc. All rights reserved. Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, and the Guidewire logo are registered trademarks, and Guidewire Insurance Suite, Guidewire BillingCenter, Guidewire ContactCenter, and Guidewire Studio are trademarks of Guidewire Software, Inc. All other product names are the trademarks of their respective companies.

This product includes information that is proprietary to Insurance Services Office, Inc (ISO). Where ISO participation is a prerequisite for use of the ISO product, use of the ISO product is limited to those jurisdictions and for those lines of insurance and services for which such customer is licensed by ISO.

This documentation is published as Guidewire Confidential. The contents of this documentation, including product architecture details and APIs, are considered confidential and are fully protected by customer licensing confidentiality agreements and signed Non-Disclosure Agreements (NDAs).

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire BillingCenter.

Version Number

- This release of Guidewire BillingCenter is 3.0.1.34
- For Standard Reporting, it supports InetSoft StyleReport Enterprise Edition 10.1, 20090924.

Support

This document includes important information about the current release of Guidewire BillingCenter. For assistance with this software release, contact Guidewire Customer Support. Contact support on the Web at <http://portal.guidewire.com>, by email at support@guidewire.com, or by phone at +1-650-356-4955.

Installing This Release

For installation information, see the *BillingCenter Installation Guide*.

InetSoft Evaluation

You can evaluate InetSoft Style Report Enterprise Edition Version 10.1 using the following license key:

L000-798-ERX-0000BF00100001F-F2AFC9DD34ED

If you want to use Inetsoft and the BillingCenter reporting module in a production environment, contact Guidewire Customer Support to request a full license key. For more information, refer to the *BillingCenter Reporting Guide* documentation.

WARNING The build number of the reporting install files must match the build number of the Guidewire BillingCenter installation. If the build numbers do not match, you will have problems with your reporting installation.

Changes in this Release

This section describes the product changes in this release.

- Display Key Changes
- Base PCF File Changes

Display Key Changes

For a list of display key changes between BillingCenter 3.0.0 and BillingCenter 3.0.1, [click here](#) (requires the `readme_files` directory on your local disk). This is a Microsoft Excel file.

Base PCF File Changes

All links below require the `readme_files` directory on your local disk.

BillingCenter release 3.0.0 to 3.0.1

- To view a report of the changes in the base PCF files in the `modules/bc` directory, [click here](#).
- To view a report of the changes in the base PCF files in the `modules/platform` directory, [click here](#).

General Issues and Improvements

The following are the improvements added to and the general issues corrected in this release:

Issue ID	Description
Accounting Subsystem	
CC-58243	Removed the Keep Extra With Policy option (<code>keepextrawithpolicy</code> or <code>tc_keepextrawithpolicy</code>) shown on the Account Distribution screen, in preparation for the deprecation of Policy Period Unapplied.
Administration	
CC-55673	Made all types of BillingCenter plans localizable on the Name/Description fields (where appropriate).
CC-55440	Exposed the parent group hierarchy to the Groups screens in the user interface (Administration → Groups → New Groups).
Agency Bill	
CC-58219	Updated the BillingCenter Plan Administration pages with fields that allow users to edit localized field for all languages in the system.
CC-58113	The following warnings in PCF configurations no longer appear when starting BillingCenter: <ul style="list-style-type: none"> WARN PaymentSearchScreen.pcf:171 Code: An implicit coercion from type <code>java.math.BigDecimal</code> to <code>java.lang.Double</code> occurs here. Consider making this explicit with an "as <code>java.lang.Double</code>". [line:6 col:26] WARN ViewAgencyDistItemsLV.pcf:99 Row: Don't use "visible" to hide Rows in a RowIterator, since it breaks paging. Filter the backing data instead.
CC-57898	A <code>HasActivePromise</code> returns a null when a payment executed as a promise is reversed. To fix this problem, added a data model field, <code>AgencyCyclePayment.AppliedFromPromise</code> . This links back to an Agency Cycle Promise if this payment was executed from a promise. If the payment was not executed from a promise, it is <code>null</code> . (There is no easy way to determine which payments are already in the system and what promise was used to execute them.) This is not a problem for those payments, unless they are reversed, in that case, the promises stays inactive. Now, when a payment is executed from a promise, this link is attached. Whenever agency payments are reversed, if this link is non <code>null</code> , BillingCenter <i>unpays</i> the associated promise. This makes the promise active again and makes all the money promised instead of paid. Until now, the way promise items were paid was to simply reverse them. However, with these new changes, it was important to distinguish the difference between a reversed promise and paid promise. Therefore, a field was added to <code>AgencyPromiseItem.Paid</code> . If this field is set to true, the promise item is currently being paid by a payment and if it is set to false, it is not. <i>Unpaying</i> promises also sets this field to false.
CC-56716	Changing the billing method from direct bill to agency bill caused information on the statement to differ from information in the New Direct Bill Payment wizard. This issue affected commission amounts and occurred when commission earnings in the producer's commission plan was set to On Binding. This issue is fixed and information is now consistent.
General Application Bug	
CC-58112	Exporting Holiday data from the Administration tab was not working. Fixed the problem so that now to export and import holidays (with their associated holiday tags) you must select Holidays from the Data to Export drop-down menu on the Export Administrative Data screen.
Assignment of Items	
CC-57642	A problem occurred in the Assign Payer popup where if invoice items were from different charges that had different payment plans then the invoice items were placed in the same invoice stream. Now, the appropriate invoice stream is used on a per-charge (and therefore per-Payment Plan) basis.
CC-57318	An exception was thrown when making a point-in-time transfer of an Invoice Item (moving commission from Producer A to Producer B), then assigning the item to Producer A's Statement Invoice. This issue has been fixed.
CC-56929	Assigning a charge to a payer and then doing a payment plan refactor (change payment schedule), caused all invoice item to revert back to the original owner To insure this does not happen, changed the user interface so now you can see the current Default Payer for every charge. Also, now the Default Payer can either be a producer or an account.
BillingCenter Integration	

Issue ID	Description
CC-57544	<p>The Renewal API fails to find prior policy period in some conversion scenarios.</p> <p>When PolicyCenter sends a new term to BillingCenter for a policy that exists in BillingCenter, but not in PolicyCenter, BillingCenter should not create a new policy period. Instead, it should find the policy and renew it.</p>
Charge/Charge Group	
CC-57394	<p>A flag has been added to a charge to skip (not create) the invoice items when received via a BillingInstruction. When this flag is TRUE, BillingCenter will not create any invoice items. By default it's set to FALSE, which means invoice items for the charge will be created.</p>
Commission	
CC-58252	<p>The ICommission plugin has a new method called <code>distributeCommissionWriteoffAcrossItemCommissions</code>. BillingCenter calls this method after any code calls <code>ChargeCommission.writeoffCommission()</code>. The plugin method must distribute the commission writeoff amount to individual ItemCommission entities under it.</p> <p>After the plugin method completes and returns the mapping of item commissions to the writeoff amount, BillingCenter validates the mapping. BillingCenter verifies the sum of the allocations are correct and then runs validation on each of the ItemCommission entities.</p>
CC-58245	<p>Commission consistency needs to distinguish between direct and agency bill.</p> <p>To resolve this issue, fixed the <code>Invoice.PaidStatus</code> consistency checks. <code>PaidStatus</code> is only related to a <code>StatementInvoice</code>'s commission, not an Account Invoice. Added a filter on <code>Invoice</code>'s subtype. Also included the <code>CmsnPayable</code> amount, because an item could have been assigned from direct bill with some payable commission.</p>
CC-57981	<p>Problem occurred where the <code>CmsnPayable</code> batch process did not move commission from reserve after an override.</p> <p>Fixed issue, so that now when commission is adjusted, it <i>catches up</i> the <code>CmsnPayable</code> to the state of the invoice item.</p>
CC-57871	<p>A database exception occurred when a commission payment was made against the producer.</p> <p>Added a user interface validation to ensure that you can no longer create a producer without giving it a primary contact first.</p>
CC-57195	<p>Changing the Commission Rate on a Policy Period through user interface from the policy level Override of the Commission Plan does not affect existing charges.</p> <p>This issue has been fixed so that now using the Override Commission Plan popup to specify an alternate commission plan will affect existing charges properly. This is done in Gosu and thus can be overridden or bypassed by customer configuration.</p>
CC-57130	Added aggregate by policy and charge filters for Item events to the Producer → Producer Statement screen.
CC-57129	Added the <code>ProducerStatement</code> method <code>findRelatedItemEvents</code> to search for item events related to the given producer statement, and changed the List View on the <code>ProducerStatement.pcf</code> to display those item events.
CC-57044	<p>Changing the invoice item amount was not changing its commission amount.</p> <p>Changing the amount on an invoice item and committing the changes now redistributes commission across planned items for every role on that charge, calculated from the item's commission rate, and its newly changed amount. Any items whose commission rates were overridden will still keep that rate, so the total commission across a charge could change when changing its items' amounts in some cases.</p>
CC-55400	Ensure that reversals always use direct bill processing to immediately earn payable and fully pay off the gross.
CC-54272	BillingCenter now allows non-commissionable items from account, collateral, and collateral requirement charges to be assigned to an agency bill payer.
CC-54167	The <code>Commission Payable</code> batch process now successfully brings commission earnings up-to-date for the <code>InvoicePaid Payable</code> criteria after items have been assigned to or from the invoice.
CC-52703	<p>In the ICommission plugin, the method signature changed for <code>shouldMakeCommissionPayable</code> so that it no longer take a charge commission (<code>ChargeCommission</code>). Instead it takes an item commission (<code>ItemCommission</code>).</p> <p>If you still want to work with the charge commission, then implement the <code>itemcommission</code> method and call <code>itemCommission.ChargeCommission</code> to get the charge commission. Finally, use that charge commission and copy your legacy behavior that worked with the charge commission.</p>

Issue ID	Description
CC-52614	<p>On the ChargeCommission entity, the <code>writeoffCommissionRecovery(BigDecimal)</code> method is now deprecated.</p> <p>In previous releases, it created a single recovery transaction. Now, it iterates over the items, and creates a writeoff or recovery writeoff depending on the sign of the allocation. Instead of this method, use <code>ChargeCommission.writeoffCommission(BigDecimal)</code> with a negative amount.</p>
CONF - Product Configuration	
CC-58261	<p>A problem occurred when creating Commission Plans if you were using a non-default locale.</p> <p>This issue has been fixed.</p>
CC-58229	Fixed localization problems where some sample data for billing, delinquency, and agency bill plans was not visible when switching to French Canadian.
CC-57397	<p>Previously, the <code>Account</code> and <code>DirectBillMoneyRcvd</code> properties on the <code>DirectBillMoneyRcvdTxn</code> were not visible from Gosu.</p> <p>This fix makes those properties visible.</p>
CC-57078	<p>Removed the <code>getAgencyPaymentItem()</code> method from <code>InvoiceItemTransactionExt.gs</code>.</p> <p>Instead, use the <code>ChargePaidFromUnapplied.PaymentItem</code> in Gosu to get the payment item.</p>
CC-57061	<p><code>PaymentMoneyReceived</code> Pre-Update rules were not triggering when posting a new direct bill payment from the user interface.</p> <p>Preupdate rules must be on the base supertype instead of the subtypes. Moved the preupdate rule from <code>PaymentMoneyReceived</code> to <code>BaseMoneyReceived</code>.</p>
CC-54381	Removed unused display keys from BillingCenter.
Document Management	
CC-57730	Fixed a problem where a null pointer exceptions occurred in the Document screen's list view when searching for a document.
CC-56944	<p>An exception occurred when trying to use the <code>EmailSent.gosu</code> document template.</p> <p>To fix this issue, modified the sample document template descriptor for a sent email to reference <code>DocumentSecurityType.confidential</code>. Prior to this, it was referencing a typekey <code>sensitive</code> that did not exist in the <code>DocumentSecurityType</code> typelist.</p>
CORE	
PL-11136	Guidewire has corrected an issue with the configuration upgrader that caused it to throw an exception and quit. This happened specifically if you were running an upgrade within a major release to a maintenance release.
PL-11068	<p>In calling a Java plugin from Gosu code, the application did not automatically convert the domain entity class to an external entity class. For the application to handle this correctly, you need to copy the external entity jar (<code>gw-entity-bc.jar</code>) from:</p> <pre>java-api/lib/ to modules/configuration/plugins/shared/lib/ to: plugins/shared/lib/</pre>
PL-10790	Guidewire has made modifications to the address-based business calendar functionality. This should greatly improve performance of these functions.
PL-10402	Guidewire has corrected an issue that if you started the tools from <code>Ant/Program.gs</code> , it produced a "NoClassDefFoundError: Files" on some Windows machines.
PL-10287	Guidewire has corrected an issue that prevented adding a column with a data type of "japaneseimperialdate" to an entity.

Issue ID	Description
PL-10183	<p>Guidewire has modified the behavior of the encryption plugin (<code>IEncryption</code>).</p> <ul style="list-style-type: none"> * It is now possible to register any number of <code>IEncryption</code> plugins, just as with messaging plugins. Only one <code>Encryption</code> plugin is the "current encryption plugin". There is a new config.xml configuration parameter called <code>CurrentEncryptionPlugin</code>. It specifies which encryption plugin (among potentially multiple implementations) is the current encryption algorithm for the main database. Set the parameter to the plugin name (not the class name). If this configuration parameter is missing, the application uses the <code>IEncryption</code> plugin with name "<code>IEncryption</code>". Any legacy existing <code>IEncryption</code> plugins have the plugin name "<code>IEncryption</code>". When you use Studio in the Plugins editor to add an implementation of <code>IEncryption</code>, Studio prompts you for a text value to use as the plugin name for this implementation. Guidewire strongly recommends you set the plugin name for encryption plugins to names that describe the algorithm. For example, "encryptDES3". * Every <code>IEncryption</code> plugin implementation returns the plugin's unique encryption ID as a String value. This is exposed to Gosu as the property <code>EncryptionIdentifier</code>. This API did not change in this release. However, starting in this release this encryption ID is very important. The application decides whether to upgrade the encryption data with a new algorithm by comparing the encryption ID of the current encryption plugin to the encryption ID associated with the database last time the server ran. If the encryption ID is different, the upgrader decrypts the data with the old algorithm and re encrypts it with the new algorithm. For new plugins set the encryption ID to describe the algorithm, for example "encryptDES3". IMPORTANT: do not confuse the encryption ID with the plugin name. The encryption ID is saved in the database. * As in previous releases, if the upgrader detects added or removed encryption for any data model fields, the upgrader encrypts the field in the main database using the current encryption plugin. * The upgrade-specific plugin interface <code>OldEncryption</code> is now unsupported and removed from the product. Its functionality is replaced by comparing the encryption ID of data in the main database with the encryption ID associated with the current encryption plugin. This triggers decryption and then reencryption of the data using the newer encryption plugin. The upgrader checks the encryption ID of the plugin to determine whether to decrypt and re encrypt the main database data.
PL-10141	<p>In calling a Java plugin from Gosu code, the application did not automatically convert the domain entity class to an external entity class. For the application to handle this correctly, you need to copy the external entity jar (<code>gw-entity-bc.jar</code>) from:</p> <pre data-bbox="425 1036 816 1151">java-api/lib/ to: modules/configuration/plugins/shared/lib/</pre>
PL-10095	<p>Corrected an issue with the <code>MessageTransport</code> plugin that caused it to not shut down if you shut down the application server.</p>
PL-10054	<p>Guidewire has modified linguistic searching in the Japanese locale and added two small Japanese letters, ? and ?, to the eight small letters already configured to be treated as equal to the corresponding full size character. This Japanese small/large letter equivalence is in effect only if you configure a linguistic search strength for the Japanese locale of "primary".</p>
PL-10027	<p>Guidewire has modified how validation works. Custom entities now trigger validation rules, as long you name the validation rule set correctly. For example, if you create an entity named <code>MyEntityExt</code>, this entity triggers validation rules in the <code>MyEntityExtValidationRules</code> rule set.</p>
PL-10002	<p>Guidewire has provided the ability to add events to an entity via extension. So, if a base entity does not declare events, a customer can add events to the entity by extending the entity and adding the <code><events></code> tag.</p>
PL-9987	<p>Guidewire has corrected an issue in which running the <code>regen-pcfmapping</code> tool did not correctly pick up custom PCF pages.</p>
PL_9914	<p>Guidewire has corrected a localization issue in which the entity type display name did not use the subtype key's localized name. This led to some user interface labels not being localized correctly.</p>
PL-9857	<p>Guidewire now provides four modes for debugging when starting the development server during configuration. You can change the debug port to use by configuring the <code>port.properties</code> file. The modes are:</p> <ul style="list-style-type: none"> • <code>dev-debug-shmem</code> (Debug dev-mode webapp using shmem) • <code>dev-debug-socket</code> (Debug dev-mode webapp using socket) • <code>dev-suspend-shmem</code> (Debug dev-mode webapp using shmem, start suspended) • <code>dev-suspend-socket</code> (Debug dev-mode webapp using socket, start suspended)
PL-9685	<p>This Japanese small/large letter equivalence will now only be in effect if a primary linguistic search strength is configured for the Japanese locale. Two small Japanese letters, ? and ?, were added to eight small letters already configured to be treated as equal to the corresponding full size character.</p>

Issue ID	Description
PL-9457	Updated the joda-time library from version 1.4 to 1.6 to correct a problem with the IBM JDK with WebSphere that returned incorrect results for class DateMidnight.
PL-9223	Guidewire now provides the ability to generate the Data Dictionary and Security Dictionary as you build the application .war file. Use the following command: • gwbc build-war -Dconfig.war.dictionary=true
PL-9163	Guidewire has modified the config.xml file to use an XML namespace, rather than a hard-coded path to a schema. The namespace is "http://guidewire.com/config". The schema is located at modules/pl/xsd/config.xsd
PL-9046	Guidewire has modified the localization.xml file to use an XML namespace, rather than a hard-coded path to a schema. The namespace is "http://guidewire.com/localization". The schema is located at modules/pl/xsd/localization.xsd
PL-8761	Corrected an issue with query generation that caused Oracle SQLException: identifier is too long. The exception occurred if the query contained an alias longer than the 30 characters allowed by the Oracle database.
PL_8746	Guidewire has corrected an issue with regen-datamapping-together and regen-datamapping-split that did not generate mappings for custom entities and extensions.
PL-8740	Guidewire has added two new methods to the IDatabaseUpgrade plugin: • function preUpgrade(context : IUpgradeContext) • function postUpgrade(context : IUpgradeContext) Use these methods as hooks to perform custom SQL before and after an upgrade. If you have already implemented IDatabaseUpgrade in order to upgrade foreign key entities to edgeForeignKey entities, then you still need to implement these new methods. However, you can simply make them non-operational.
Equity Dating	
CC-56490	A problem occurred with policy equity calculation where the calculation could be wrong if the policy was created prior to Policy Effective Date and the policy issuance billing instruction contained more than one charge. This issue has been fixed.
History	
CC-57353	Replaced the Producer → Agency Bill Cycles → Agency Bill Item History screen/Transactions List View with the Producer → Agency Bill Cycles → ItemEventPanelSet.
CC-57127	Create the Item History screen. Added an IInvoiceItemDetailPopup which contains a list view that lists all events associated with the item.
CC-57126	Associated (hooked) Item history to the events that trigger them. There is an addItemEvent method on InvoiceItem, which takes an event type and a transaction, if one exists, and creates an item event associated with the invoice item. If the transaction is of type producer transaction, associate the item event with the corresponding producer code.
Holds	
CC-57508	When a billed invoice has a charge that gets held after running the Invoice Due batch process, create billed transactions for the offset if it is placed on the current invoice.
CC-44581	Held account-level charges are not showing up on the Desktop → Held Charges screen. Fixed the Account column on Desktop → Held Charges screen so that it now shows the correct account charges.
General Integration	
CC-57881	Created a new IDateSequence plugin for customizing the creation of date sequences. Moved the CreatePeriodicSequenceWith method from the IPaymentPlan plugin to the IDateSequence plugin. Also, enabled the DateSequence.createTwicePerMonthDateSequence method to use the plugin to create the two monthly date sequences.
CC-57860	Changed the PaymentPlan plugin so that now each of the four methods are now passed the charge as the context for the invoice items being created. Previously, the charge was not one of the parameters.
CC-	The IDataExtractionAPI web service is now deprecated, as described in the What's New and Changed chapter for integration changes. In addition, note that the web service no longer has the @WebService annotation. If you need to use this temporarily to transition your integration code to custom services you can do so by removing the comment marker before the annotation. However, Guidewire strongly recommends against using the methods that have the @DoNotPublish annotation. Those methods allow code injections, so you must not use them in a production system.

Issue ID	Description
Gosu	
PL-9692	Corrected an issue that caused AuthenticationSourceCreatorPlugin implemented in Gosu to fail authentication if used with WebSphere 7.
PL-9275	Studio now generates deprecation reason messages for MetaType properties exposed from TypeLiteral expressions.
Integration	
PL-10629	Guidewire has modified the way it supports InetSoft VPM. This work has tightened the Guidewire SOAP security for ISREEAAuthenticationAPI and added three more properties to sree.properties. <ul style="list-style-type: none"> • gw.soapuser.username: The user name used by InetSoft to retrieve user/group/role information used by VPM and authentication • gw.soapuser.userpassword: The password associated with the user name. • gw.usercache.size: The cache size for user cache, group cache, and role cache.
PL-9900	Guidewire has corrected an issue with incorrect formatting and parsing of the values for XSD gMonth, gDay and gMonthDay.
PL-9848	Guidewire has fixed the PCF for Holiday, businessweek, catastrophe configuration so that all the zones with granularity configured will show up in the selection.
PL-9721	Guidewire corrected an issue in which disabling the ActiveX document control caused an attempt to create a document from a template to fail. You can now perform this task even if you disable this ActiveX control.
PL-9714	Guidewire has deprecated the following method: <code>gw.api.webservice.zone.ZoneImportHelper.importDefaultDataToProduction</code> which deprecated the support for the "dataFile" attribute in zone-config.xml
PL-9659	Guidewire has modified the base configuration address-config.xml entry for Canada so that autofill works correctly for both city and postal code.
PL-9569	Guidewire has corrected an issue with reporting JAR file (gw-sree.jar). Previous builds did not automatically contain the Security Provider classes.
PL-9353	Part of the Guidewire/InetSoft reporting integration involves a security plugin. Guidewire has modified the plugin implementation to now cache up to one user's information to reduce the number of soap calls made between a Guidewire application and the integrated InetSoft server. The cache has a default timeout of 30 seconds. You can modify this value by altering the property "gw.usercache.timeoutinseconds" in sree.properties. The value is in terms of seconds.
PL-7931	Guidewire now supports the use of WebLogic as the Report Server.
INVC Charge/Invoicing	
CC-58285	Remove Periodicity.FULL_PAY from the product.
CC-58047	Validate that the total amount of all Invoice Items on a charge equals the charge's total amount. To fix this issue added a preupdate and insert validation that verifies that the amounts of the invoice items on a charge adds up to the charge amount.
CC-57941	There is a new plugin called <code>IInvoiceStream</code> that allows you to customize how BillingCenter chooses the right invoice stream based on the account, the charge, and the payment plan. The default plugin implementation just uses the default invoice stream selection based on the periodicity of the payment plan argument to the plugin methods. For more details, see the Integration Guide in the "Billing Plugin" topic.

Issue ID	Description
CC-57907	<p>The DateRecurrence class has been removed, and replaced by the DateSequence.</p> <p>The timing of recurring producer payments was specified with a RecurrenceFactor (such as 'every') + RecurrenceBasis (such as monthly) + payment day of month. Now, this is specified as a Periodicity typecode (such as 'monthly') + payment day of month.</p> <p>However, a producer still has a ProducerPaymentRecurrable that pays a producer's commissions on a recurring basis. From now on, the responsibility for knowing the frequency of payments moved from the RecurrencePattern to the Recurrable. The ProducerPaymentRecurrable entity does contain a RecurrencePattern but only a Recurrable. The Recurrable has a new property (Periodicity) to contain the Periodicity typecode. The RecurrableImpl creates the date sequence by its Periodicity typecode and then calling out to the calling out to the DateSequence plugin.</p> <p>ProRataCharge previously specified the timing of its earn dates by implementing entity RecurrencePattern, which held a RecurrenceFactor (such as 'every') and RecurrenceBasis (such as monthly). Now, this behavior is replaced with a new Periodicity attribute (such as 'monthly') on ProRataCharge.</p> <p>As part of this change, there was a change to the web service interface IBillingCenterAPI. In the createProducer method, the parameters for the producerPaymentFrequency and producerPaymentBasis are replaced by one parameter for producerPaymentPeriodicity.</p> <p>Also, in the PaymentPlan entity, the method createRecurrence is now deprecated.</p>
Cc-57900	<p>Problem occurred where running the Invoice batch process.</p> <p>To fix this issue, made changes to the Invoice Distributed Workers.</p>
CC-57757	<p>When creating a new invoice and changing the invoice send date such that the lead time is different than the billing plan setting, BillingCenter created a new empty invoice.</p> <p>Before this, using a custom Invoice Assembler plugin could result in new empty invoices being saved to the database. If the Invoice Assembler plugin created a new invoice for an item and then decided not to put any items on that invoice, it would still save that empty invoice to the database.</p> <p>Now, the Invoice Assembler will never save a newly created invoice to the database if it is empty. (This applies to both account invoices and statement invoices.)</p>
CC-57722	<p>There is a new plugin called the Invoice Assembler plugin (IInvoiceAssembler). Use it to customize how BillingCenter maps bill invoice items on to invoices.</p> <p>Refer to the Integration Guide in the "Billing Plugins" topic for details.</p>
CC-57576	<p>To customize the date sequence for all invoice streams based on a specific periodicity, modify the DateSequence plugin's createPeriodicSequenceWith() method. Return an implementation of See the Integration Guide in the "Billing Plugins" chapter for details. As a consequence of this change, DateRecurrence and its subclasses are removed.</p>
CC-57505	<p>Added the method InvoiceStream.createAndAddInvoice(Date sendDate) that creates and adds an invoice to this stream to be sent (billed) on the given date</p>
CC-57449	<p>The InvoiceAssembler treated invoices for accounts that had a fixed due day of month as <i>lookahead</i>, meaning they could take invoice items whose dates were any time before the due date of the invoice. Invoices for accounts that had a fixed bill day of month did not have <i>lookahead</i> so they could take only invoice items whose dates were before the invoice bill (event) date.</p> <p>Now, the InvoiceAssembler places an item on the earliest invoice whose Due Date is after the item's date if the item is using Due Date placement. It places the item on the earliest invoice whose event(Bill)Date is after the item's date if the item is using Bill Date placement.</p>
CC-57398	<p>Adding an invoice item to the Account → Charges screen and then immediately removing it while not clicking Update in-between, caused a null pointer exception.</p> <p>This issue has been fixed.</p>
CC-57249	<p>On the InvoiceItemDetailPopup added columns to the ItemEvent List View which detail how the event affected the gross and commission on the item.</p>
CC-57065	<p>Previously, the user was not allowed to set an invoice item's amount to have the opposite sign of its corresponding charge's amount.</p> <p>However, that restriction has now been removed.</p>
CC-56862	<p>Remove the InvoiceItemType DEPOSIT_SEPARATE and all code that had special cases for the DEPOSIT_SEPARATE type invoice items from the product.</p>
CC-48668	<p>Added the ability to change the invoice date to be equal to today's date.</p>

Issue ID	Description
CC-48069	Added a new field to the Charge entity called Charge Group (Type String).
Letter of Credit	
CC-57622	Fixed the Legacy Letter Of Credit batch process to make it more generic and to look for all collateral requirements.
Manageability	
PL-10110	Guidewire has modified the Server Tools → Work Queue Info page to show additional worker summary information, including throughput information.
PL-10000	Guidewire has added two additional columns to the Index Physical Statistics tab of the Server Tools → Database Storage Information page. The two new columns are fill_factor and is_padded.
PL-9911	Guidewire has modified the Server Tools → Database Storage page to include the following information: <ul style="list-style-type: none"> • Queries Executed to Build Download • Summary of Queries Executed to Build Download
PL-9812	Guidewire has modified the Server Tools → Database Storage page to allow you to choose choosing specific tables and the mode.
PL-9697	Guidewire has modified the Server Tools → Data Distribution page so that you can generate data on the minimum and maximum array sizes that shows the row count changes across distributions. This is for all arrays in the data distribution download.
PL-9385	Guidewire has corrected an issue that generated a Null Pointer Exception if you clicked the Download button in the (Server Tools) Workflow Info page. The exception occurred if there were any work items that had a status of Error or Failure.
PL-8405	Guidewire has added new Oracle-only parameters to the System Tools → Database Parameters Info page. They are: <ul style="list-style-type: none"> • Linguistic Search Options • Linguistic search strength • Linguistic search function name
PL-8207	Guidewire has modified the colors used on the charts in the System Tools → Cache Info page to make the information more clear.
PL-8108	Guidewire has implemented database searching and sorting appropriate to non-English locales.
PL-8055	Guidewire has added a new download to the System Tools → Data Distribution Info page that reports row counts and growth rates of tables within a specified span.
Payments	
CC-58326	Changing item amounts did not modify payment items made against them. Now, when an invoice item's amount is changed, all direct bill payments against the charge is reversed and the amount that was reversed on each payment is redistributed to the charge. This ensures that commission earnings work correctly.
CC-58120	Modified the AdministrationPaymentPlan.pcf in the user interface to remove unused Card Panels.
CC-57869	An exception occurred after calling distItem.reverse() and trying to execute the modified payment. To fix this issue, moved the reversing logic from the charge paid from unapplied transaction to the payment item. Also added a removeOrReverse() method on BaseDistItem so executed items are not removed by default.
CC-57942	Ad Hoc statements (such as statements off the normal cycle) are supported in agency bill processing so they can be added to invoice streams.
CC-57143	Added a new reversed dist items tab to the Account → Actions → New Direct Bill Payment screen's payment detail to show reversed items. The tab is only visible if there are reversed items.
CC-57141	Added Distributed Amount and Amount in Suspense columns to Account → Actions → New Direct Bill Payments List View.
CC-57096	Fixed the AgencySuspensePaymentWorkQueue (batch process) process so that it only handles agency bill suspense items. It should not query for direct bill suspense items.
CC-57095	Provided the capability to modify the amount or policy number of a suspense item from the Account → Actions → New Direct Bill Payments screen.

Issue ID	Description
CC-57092	<p>Added suspense items to direct bill payments. The direct bill suspense items behave the same way as agency bill suspense items except that there is no auto-release of these items. The amount is held in suspense for a policy period until it is manually released and then the customer can apply it appropriately.</p>
CC-56667	<p>Direct Bill Suspense Items have been implemented to handle the following use case.</p> <p>Suppose that a payment comes into BillingCenter, and some or all of that payment is targeted at charges of a policy that does not yet exist in BillingCenter. (For example the policy might exist in a policy administration system, but has not yet been created within the BillingCenter system)</p> <p>In such a situation, users can create a suspense item on the direct bill payment screen to hold funds until the policy actually arrives in the system. When creating the suspense item, the user enters the name of the policy, the amount of the payment that should target the policy, and a set of optional comments.</p> <p>When the payment is executed, the system places the amount that was specified for the suspense payment into a Suspense T-account on the Account.</p> <p>Once users want to release the funds from the Suspense T-account and apply them (for example, once the policy period in question actually does exist within the system), they can edit the direct bill payment they executed earlier and specify that they want to release the suspense item. This frees the suspended funds and they then can be used to target invoice items from the policy that is now in the system.</p> <p>NOTE TO CONFIGURATORS: As part of the work for this feature, <code>ModifyDirectBillPayment.pcf</code> was removed and replaced with <code>ModifyDirectBillPaymentPopup.pcf</code>. So anywhere in configuration that you were using <code>ModifyDirectBillPayment.pcf</code>, now use <code>ModifyDirectBillPaymentPopup.pcf</code>.</p>
CC-56583	<p>Prior to this fix, when a payment was made against a set of invoice items associated with a policy period and then a payment plan change was made to the policy period, the payment distribution items made against any invoice items removed/reversed as a result of the payment plan change would themselves be reversed and this would put money into Account Unapplied. However, the money should then have been redistributed against remaining and newly created invoice items, and this was not happening.</p> <p>Now, as part of the payment plan change process, payment distribution items that were targeting invoice items which were removed/reversed by the payment plan change are reversed, and money that was pushed into account unapplied as a result is redistributed so that it pays invoice items (and does not just sit in account unapplied).</p> <p>A new plugin method was added to the <code>DirectBillPayment</code> plugin that gets called as part of this process:</p> <pre>allocateForRedistribution(payment: ModifyingDirectPayment, chargeToAmountMap : Map<Charge, BigDecimal>)</pre> <p>This method is passed a payment which has had some or all of its distribution to invoice items reversed as part of the payment plan change, and it is also passed a Map that contains a set of (Charge, BigDecimal) ordered pairs that indicate how much of the payment's distribution was reversed for each charge. A default Gosu implementation for this method is provided but users can make their own changes as needed.</p>
CC-55527	<p>Remove the following Policy Payment entities:</p> <ul style="list-style-type: none"> • <code>PolicyPayment</code> • <code>AgencyPolicyPayment</code> • <code>AbstractrPolicyPayment</code>
CC-55239	Enabled the View Direct Bill Payments Screen to show ZeroDollarDMRs in a separate tab.
CC-54969	<p>Changed handling of point-in-time commission transfers to a Null producer. It calls to a new plugin method: <code>ICommission.shouldWriteoffCommissionWhenProducerCodeRemoved(ItemCommission)</code>, which decides what to do with existing commission when users transfer an item from a producer to a Null producer. They can either:</p> <ul style="list-style-type: none"> - Writeoff the remaining reserve on the producer. - Override commission rate on the producer to 0%, which will un-earn anything that was payable or paid (perhaps not immediately).
CC-53398	<p>Paying a Billed negative item that is not due makes the policy period Past Due.</p> <p>When negative invoice items are billed, the credit is not taken unless there is an offsetting positive charge to apply it to. If there is any remaining amount on the credit, it will be taken when the invoice goes due.</p> <p>There is now an <code>AllocateCreditsDistribution</code> strategy which is called by default when the <code>DirectBillPayment.allocateCredits()</code> plugin point for <code>InvoiceBilled</code> context is called.</p> <p>Removed the negative items context and note the actual context that accessed the negative items (such as: <code>Invoice Billed</code>, <code>Invoice Due</code>)</p>

Payment Plan and Invoicing

Issue ID	Description
CC-57944	<p>Added the Name property to InvoiceStream and displayed it in the user interface on the Agency Bill Cycles and account's Invoices screens.</p> <ol style="list-style-type: none"> 1. Added a column called Name to InvoiceStream as a non-nullable field. The default name is Default. 2. Added a new column to both the AccountDetailInvoices.pcf and ProducerAgencyBillCycles screen. 3. Modified AccountImpl to change the InvoiceStream's name to the Base Periodicity.
CC-57579	<p>The payment plan offset days are not being respected.</p> <p>Updated the InvoiceAssembler.gs code to use the methods on invoice stream instead of the builder.</p>
CC-57512	<p>Changing the payment plan on an agency bill policy sometimes fails. This can happen if a payment already existed on the producer and the payment contained multiple items, at least one of which was on a Planned cycle and at least one of which was on a Billed cycle.</p> <p>This issue has been fixed.</p>
Producer	
CC-57936	<p>Added a findRelatedItemEvents domain method to the Producer Statement.</p> <p>The findRelatedItemEvents domain method finds item events with the item event types of Earned, Commission Moved to, and Commission Moved From. It gets the items related to producer transactions that are related to a given to the Producer Statement.</p>
CC-57931	<p>After a retroactive producer transfer the transaction reversals were incorrectly being placed on an existing statements.</p> <p>Fixed this problem so that now transaction reversals are placed on a new statement the next time the Producer Payment batch process runs.</p> <p>Reversed producer transactions do not automatically go on an existing producer statement</p>
Renewal Process	
CC-56691	<p>The Closure status and earned premium amount incorrectly populated the new term of a policy period when an insured first closed the policy and then renewed it.</p> <p>Fixed problem where the Closure status was not being properly set upon renewal.</p>
Reporting Subsystem	
CC-58165	<p>The Commission Aging reports were not showing commission aging correctly for commission payments.</p> <p>The reports now correctly reflect the amount based on transaction dates.</p>
CC-57932	<p>Problem occurred where the Writeoff report was showing incorrect policy premium writeoff amounts because duplicate records were being created at the invoice and invoice item levels.</p> <p>To fix this issue, changed the approach so that it now goes through the transaction to the Account T-account.</p>
SECU (Security (Login, Permissions, Roles)	
CC-52166	<p>Added the following three new permissions to BillingCenter that allow a user to override the security zone check for viewing accounts, policy periods, and producers respectively (one permission for each of those entity types):</p> <ul style="list-style-type: none"> • acctignoresecurityzone • plcyignoresecurityzone • prodignoresecurityzone <p>By default, these three permissions are now contained in the superuser role.</p>
Staging Tables	
CC-57624	<p>Added a StatementInvoiceItemLoaderCallback to populate the StatementInvoiceItem.promisedAndPaidAmount and the StatementInvoiceItem.promisedAndPaidCommission columns for the InvoiceItem table.</p> <p>Made the ItemState.PromisedAndPaidAmount and PromisedAndPaidCommission fields have the attribute overwrittenInStagingTables="true".</p>
CC-57620	<p>Added AgencyCycleDistLoaderCallback to populate the AgencyCycleDist.NetAmountDistributed column.</p> <p>For each AgencyCyclePayment row, sets the netAmountDistributed to be the sum of grossAmountToApply - commissionAmountToApply for the payment's agencyPaymentItems.</p> <p>It does not populate the column for AgencyCyclePromises because promises are not loadable.</p>

Issue ID	Description
CC-57617	Agency Bill writeoffs now correctly calculate the unpaid gross amount and unpaid commission amount during a staging table load, and persist these values as denormalized fields Amount, CommissionAmount on the Writeoff.
Studio	
PL-10369	Corrected an issue in which Studio did not properly recognize the VALUE keyword conversionExpression attribute (in a PCF page) causing Studio to display an error message.
PL-10363	Corrected an issue in which the Studio global replace functionality did not work correctly within rule set categories, generating an error message.
PL-10357	Corrected an issue that prevented Studio SmartHelp from providing a selection list for Group or User ID if creating assignment rules.
PL-10333	Guidewire has corrected an issue with the Revert to Base right-click command on PCF folders deleting the base configuration folder.
PL-1-329	Corrected a Studio search option that did not work correctly if searching from the lowest node in the rule set hierarchy. You can no longer search on Rule Set View in the global replace dialog.
PL-10293	Guidewire has corrected an issue that caused a Null Pointer Error if you double-clicked a view tab to maximize it.
PL-10185	Guidewire has corrected a Studio issue with searching that threw an error if you were searching within a rule set containing CVS or Subversion files. This issue only occurred if you searched for text or names within the rule resources, not with any other Studio resources.
PL-10020	Guidewire has corrected an issue in which the Studio Rules editor did not properly escape double quote marks in the rule description. This caused rules to not behave as expected and caused Studio to not start upon attempting to restart it.
PL-9774	Guidewire has added the ability to change the root entity of a rule. To do so, select a rule set in Studio, right-click, and select Change Root Entity from the drop-down menu.
PL-9666	Corrected an issued in which implementing the OldEncryption plugin caused Studio to incorrectly show the plugin under Plugins → gw → plugin → util → OldEncryption as well as Plugins → gw → plugin → util → IEncryption → OldEncryption.
PL-9214	<p>It is possible to configure Studio to open .xml files directly in an XML editor that is external to Guidewire Studio. To facilitate this process, Guidewire now provides the following optional XML attribute:</p> <pre>xmlns="http://guidewire.com/datamodel"</pre> <p>This attribute defines the namespace for the XML elements in the file. For information on how to use this feature, see "Working with an External XML Editor" in the Configuration Guide.</p>
PL-8997	Corrected an issue in which Studio failed to correctly create a new Rule Set category in the Resources tree if no rules directory existed in the configuration module.
PL-437	Guidewire has changed Studio behavior so that it now flags an empty Rule Condition as an error and displays a red error marker in the right-hand margin of the Condition editor. If you verify the rule, Studio generates a "Missing return value" error.
Suspense Payment	
CC-57732	<p>A MultipleMatches Exception occurred when a suspense payment targeted a policy with multiple policy periods.</p> <p>The default implementation of the SuspensePayment plugin method assumed that only one policy period existed on a policy when a suspense payment was applied to a policy period. It called <code>getAtMostOneRow</code> on a query that could return multiple items.</p> <p>To fix this issue, the method has been changed so that if more than once policy period exists, the payment is applied to the most recent one.</p>

Issue ID	Description
CC-57177	<p>Previously, if the user followed the sequence of events a suspense release transaction would be issued for an incorrect amount:</p> <ol style="list-style-type: none"> 4. Create a Direct Bill or Agency Bill Payment with a suspense item in it 5. Execute the payment 6. Modify the payment by changing the amount of the suspense item 7. Execute the payment again 8. Modify the payment by releasing the suspense item 9. Execute the payment again <p>This sequence of events resulted in a release transaction being issued for the original amount of the unmodified suspense item, not the updated amount.</p> <p>Now, the release transaction is issued for the correct amount, which is the amount of the modified suspense item at the time it is released.</p>
Trouble Tickets	
CC-56416	<p>Cancelling the creation of a trouble ticket in the middle of the wizard causes an Eval Exception to occur.</p> <p>To fix this issue, modified the CreateTroubleTicketWizard.pcf function that is called in the wizard's afterCancel method so that if you cancel the trouble ticket during a non quick jump situation.</p>
Web	
PL-10130	Guidewire has corrected an issue in which a multi-select input did not correctly display the selected items in read-only mode.
PL-9396	Corrected an issue that caused the menu button label color to be incorrect if user selected a list row.
PL-9328	Fixed an issue in which a CSV export to Microsoft Excel using an English (UK) locale created an invalid character.
PL-8749	Guidewire has added a new configuration parameter: WizardPrevNextButtonsVisible. Setting this parameter to true always renders the Back/Next buttons on a wizard step even if is not available (rendered as grayed-out), to maintain consistent positions of all other buttons in the wizard toolbar.

Known Issues and Limitations

Note: For maintenance releases, it is Guidewire policy to defer fixing configuration issues if they require merging files during the upgrade. Workarounds to many of these issues are listed in the following sections. The goal of this policy is to make upgrades as straightforward as possible.

This section describes known issues with this release of Guidewire BillingCenter.

BillingCenter 3.0.1 does not integrate with PolicyCenter 4.0.1 or 4.0.0 without the following changes

Issue: BillingCenter 3.0.1 is configured to work with PolicyCenter 4.0.2. However, this version is not available. To integrate BillingCenter 3.0.1 with Policy Center either 4.0.1 or 4.0.0, you must complete the steps shown in the Workaround section.

Workaround: Make the following changes to your PolicyCenter installation:

1. In PolicyInfoExt.gsx, change:

The syncBasicPolicyInfo method from: this.PeriodID = period.PeriodId.Value to TermNumber = period.TermNumber

2. In BCBillingSystemPlugin.gs, change:

- The renewPolicyPeriod method from renewalInfo.PriorPeriodID = period.BasedOn.PeriodId.Value to renewalInfo.PriorTermNumber = period.BasedOn.TermNumber
- The rewritePolicyPeriod method from renewalInfo.PriorPeriodID = period.BasedOn.PeriodId.Value to renewalInfo.PriorTermNumber = period.BasedOn.TermNumber

- The `waiveFinalAudit` method from `policyPeriodInfo.PeriodID = period.PeriodId.Value` to `policyPeriodInfo.TermNumber = period.TermNumber`
 - The `scheduleFinalAudit` method from `policyPeriodInfo.PeriodID = period.PeriodId.Value` to `policyPeriodInfo.TermNumber = period.TermNumber`
 - The `getPeriodInfo` method from `policyPeriodInfo.PeriodID = period.PeriodId.Value` to `policyPeriodInfo.TermNumber = period.TermNumber`
 - The `transferPolicyPeriods` method from `:PeriodID = p.PeriodId.Value` to `:TermNumber = p.TermNumber`
3. Start the BillingCenter server.
4. From PolicyCenter, refresh the `BCBillingSystemAPI` and `BCBillingSummaryAPI` WSDLs (Web Services).
5. From BillingCenter, import the Sample Data.
6. From PolicyCenter, import the Sample Data
7. Verify that the producers from PolicyCenter are pushed to BillingCenter.
8. Verify that a policy created in PolicyCenter appears in BillingCenter.

[Performance problems with business date utilities \(PL-10790\)](#)

Issue: BillingCenter can experience extensive performance problems during calculations involving business date functions.

Workaround: Guidewire is aware of this issue.

[Possible name space collision with web services \(PL-10599\)](#)

Issue: It is possible to experience namespace collisions in the names of argument types and return types on published web services.

Workaround: Guidewire is aware of this issue.

[Zone data cache loading performs badly and can crash the application server \(PL-10874\)](#)

Issue: Handling large amounts of zone data can cause a severer performance bottleneck. This has the potential to crash the application server.

Workaround: Guidewire is aware of this issue.

[Two new default reports will not be included in this release \(CC-55591\)](#)

Issue: The two new Guidewire reports: Account Balance and Producer Balance were not made available in this release.

Workaround: Guidewire is aware of this issue.

[WebSphere timeout during initial startup \(CC-4904\)](#)

Issue: The initial creation of the BillingCenter database may take longer than the default WebSphere startup timeouts will allow, and WebSphere may abort the startup before the database upgrade has completed. In this situation, the `Default_Server_stderr.log` file will show an interruption in the “Applying upgrade step ### of #####” log messages.

Workaround: Increase the Ping Timeout and Ping Initial Timeout values for WebSphere. These can be found in the WSADMIN Console; click **Nodes**, drill down to your application server, click the **Advanced** tab, and increase the values.

Acrobat Sample document template does not allow extra fields (PL-1416)

Issue: Currently, the example Acrobat document template uses two fields, and this requires the same number of fields in `SampleAcrobat.pdf.descriptor` and `SampleAcrobat.pdf` files.

Workaround: To allow more fields in the template and documents created from it, you must update its descriptor file.

Incorrect additivity statements in the default logging.properties file (CC-50615)

Issue: Some additivity statements in the `logging.properties` file are incorrect and therefore, ineffective. An additivity statement attempts to keep the category from sending its log messages up to its parent, preventing duplicate log messages across files.

Workaround: In the `logging.properties` file (located at `modules/cc/config/logging`), change:

```
log4j.additivity.PluginsLog=false  
to  
log4j.additivity.Plugin=false
```

BillingCenter login remains active in new browser window (CC-4152)

Issue: When you log into BillingCenter in a browser window, then you close that window and open a new one, in some instances you can access BillingCenter pages without needing to log in again.

This is an issue with some versions of Microsoft Internet Explorer. When multiple browser windows are running in the same process, they can share session information, and this allows them to share a single login with BillingCenter. Early versions of Internet Explorer provided a **Launch browser windows in a separate process** option, which, if enabled, would avoid this situation. Some versions of Internet Explorer do not allow you to set this option, and enable or disable it automatically. You can review the relevant Microsoft Knowledge Base Article 240928. However, you may experience a situation in which Internet Explorer does not correctly enable this option, especially if you manually disabled it in an earlier version and then later upgraded. In this case, the multiple browser windows are run in the same process, exhibiting the undesired behavior.

Workaround: The simplest solution is to always click **Logout** at the top of the BillingCenter window before closing your browser.

If desired, you can also take steps to manually enable the Internet Explorer option. This requires you to edit the Windows registry on the affected computer.

WARNING Editing the registry is a risky procedure, and any improper changes may result in permanent data loss and require Microsoft Windows to be completely reinstalled on that computer. To make this change, create or edit the

`[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\BrowseNewProcess]` registry entry. Set the data type to `REG_SZ`, and then set the data value to `yes`.

WebSphere timeout during initial startup (CC-4904)

Issue: The initial creation of the BillingCenter database may take longer than the default WebSphere startup timeouts will allow, and WebSphere may abort the startup before the database upgrade has completed. In this situation, the `Default_Server_stderr.log` file shows an interruption in the `Applying upgrade step #### of #####` log messages.

Workaround: Increase the Ping Timeout and Ping Initial Timeout values for WebSphere. These can be found in the WSADMIN Console. Click **Nodes**, drill down to your application server, click the **Advanced** tab, and increase the values.

BillingCenter displays many warnings when regenerating Javadoc (CC-55564)

Issue: This issue occurs when running the `regen-java-api` and `regen-soap-api` command.

Workaround: Guidewire is aware of this issue.

On AIX without an X11 server, BillingCenter may report that an X11 resource is missing (CC-5889)

Issue: BillingCenter does not require an X11 server; however other components of the BillingCenter environment (such as Java) may inadvertently have this requirement. If so, BillingCenter may report that an X11 resource is missing when it is starting up.

Workaround: Install an X11 server, or contact Guidewire support to determine if the dependency can be removed.

Port conflict when starting BillingCenter with JMX RMI adapter enabled and Microsoft Outlook running (CC-6964)

Issue: When Microsoft Outlook starts, it chooses several ports to use for its connection to an Exchange server. In some instances, Outlook may use port 1099, which is a port that the BillingCenter JMX RMI adapter can use, *if enabled*. The BillingCenter server then reports a port conflict during startup when it tries to use that port for the RMI adapter.

Workaround: Close Outlook to free the port, and then start the BillingCenter server again. You can also change the port that the RMI adapter uses by editing the `JMXManagementPlugin` through Guidewire Studio.

Studio PCF Editor treats commented-out property values as errors (PL-4582)

Issue: If you attempt to comment out a property value in the Studio PCF editor, Studio treats this as an error.

Workaround: None. Guidewire is aware of this issue.

Incomplete implementation of non-English locale-specific database search and sort (PL-8108)

Issue: Guidewire has begun implementation of database searching and sorting appropriate to non-English locales. This release of BillingCenter contains an incomplete implementation of this work. The changes will not affect customers implementing case-insensitive searches using standard English sort criteria.

As part of this work, the current release renames the `supportsCaseInsensitiveSearch` attribute on column to `supportsLinguisticSearch`.

Workaround: Guidewire is aware of this issue.

Creating Standard XSDs and Standard XML documentation is missing from the HTML documentation (CC-58429)

Issue: BillingCenter exports business data entities, Gosu class data, and other types to a standard Guidewire XML format, using a feature called GXTools. Due to a processing error, the documentation for this feature is not available in the HTML version of the documentation.

Workaround: However, you can read about this feature in the PDF version of the documentation. See Gosu Reference Guide → Gosu and XML → Creating Standard XSDs and Standard XML.

Server exception during startup (PL-8167)

Issue: During server startup, there can be a Tomcat server exception. However, after the exception, the server does start up. This is caused by the Tomcat application server attempting to either save or restore sessions using serialization. The session restore causes an exception because BillingCenter sessions contain objects which rely on the metadata being started up.

Workaround: None. The Tomcat application server does start up after the error.

[Zone data with missing values cannot be loaded \(PL-9123\)](#)

Issue: Zone data, which is used for associating postal codes with cities and regions, does not load if it is incomplete.

Workaround: Guidewire is aware of this issue.

[H2 development database creates LONGTEXT fields as VARCHAR\(65000\), instead of as CLOB as in Oracle and SQL Server \(PL-9314\)](#)

Issue: Guidewire creates schema fields defined as LONGTEXT as LONGVARCHAR columns, instead as CLOB as in the Oracle or SQL Server databases. In addition to the inconsistency involved, this mandates a field length of 65,000 characters or less in LONGTEXT columns in the H2 database.

Workaround: Define a LONGTEXT field as a VARCHAR(...) with the required size, for example, as VARCHAR(120000).

[Report Folders Do Not Display If Empty \(PL-1281\)](#)

Issue: Guidewire BillingCenter does not display a report folder if that folder is empty.

Workaround: Place an InetSoft replet inside the folder. You can mark the replet as *not visible* within the InetSoft Enterprise Manager. This ensures that the parent folder displays in BillingCenter even if the folder is empty.

[Changes to encryption column can cause encrypted columns to shrink \(PL-9665\)](#)

Issue: If you change the encryption algorithm in such a way that the encrypted columns shrink, then it causes the resulting upgrade to crash on SQLServer.

Workaround: Guidewire is aware of this issue.

[BillingCenter does not display empty report folders \(PL-1281\)](#)

Issue: BillingCenter does not display a report folder in the **Administration → Report Admin** page if that report folder is empty.

Workaround: Within InetSoft Enterprise Manager, insert a replet into the empty report folder. You can then mark this replet as not visible (using the InetSoft Enterprise Manager), which causes the replet to not show in BillingCenter. However, BillingCenter does display the parent folder.

[Report Folders Do Not Display If Empty \(PL-1281\)](#)

Issue: Guidewire BillingCenter does not display a report folder if that folder is empty.

Workaround: Place an InetSoft replet inside the folder. You can mark the replet as *not visible* within the InetSoft Enterprise Manager. This ensures that the parent folder displays in BillingCenter even if the folder is empty.

[Not Possible to Hot-Deploy on all Supported Application Servers \(PL-9955\)](#)

Issue: It is not possible to hot-deploy on all supported application servers.

Workaround: If you use Tomcat, you are able to see the changes without requiring a restart of the application server. If you use an other application server, you need to perform a restart to see any configuration changes reflected.

Values in MIME-Type Drop-Down Are Not Localized (PL-10921)

Issue: BillingCenter does not localize the values in the **MIME type** drop-down list properly.

Workaround: If you wish to localize MIME types, you can do so by modifying the definition in `config.xml`. Currently, BillingCenter does not support MIME type definitions in multiple languages.

Cancel Button Does Not Work Properly (PL-10926)

Issue: The **Cancel** button does not work correctly if you start in edit mode by clicking on a button such as **Add** from read-only mode.

Workaround: Use the **Edit** button to move the page into edit mode. In this case, all behavior is as expected.

Finder in Annotation Causes Throws Error (PL-10812)

Issue: Studio throws error and fails to do syntax error highlight, code completion, and similar functions, if there is a finder in an annotation. This occurs even if you are connected to the server.

Workaround: Remove the finder from the annotation.

Navigating Between Pages in Wizard Loses Data (PL-10920)

Issue: In a wizard pop-ups, it is possible to lose data if you navigate to another page in the wizard.

Workaround: Navigate out of the wizard and then back into it rather than changing steps directly in the wizard.

Studio Does Not Properly Report Verification Error (PL-10822)

Issue: Studio does not properly report a verification error if a typecode contains a trailing space.

Workaround: Remove trailing spaces from all type codes.

Double Quotes in Path Environment Variable Causes Problems (PL-10981)

Issue: Guidewire BillingCenter does not run if there are double quotes in the **PATH** environment variable.

Workaround: Check the **PATH** variable and remove any double quotes.

Studio Debugger Does Not Handle Rule Breakpoints Properly (PL-10553)

Issue: If you set a breakpoint in a Gosu rule and attempt to step through the code using the Studio debugger, the debugger does not correctly display the code on the breakpoint step.

Workaround: Guidewire is aware of this issue.

Inability to Access Typecodes that Contain a Dash (PL-10532)

Issue: It is not possible to access a typekey code (typecode) that contains a dash using Gosu.

Workaround: Remove any dashes from the typekey codes.

Method assignByUserAttributes Does Not Distinguish Between Attributes (PL-10607)

Issue: Method `assignByUserAttributes` does not distinguish between different attributes.

Workaround: Ensure that you thoroughly test all assignment scenarios to ensure that assignment happens correctly.

[Incorrect totals in the policy payment schedule occurred on agency bill statements after a change was made to one of two producers on a policy \(CC-56925\)](#)

Issue: The producer statement does not show the correct dollar amount after a change is made to one of the policy's payment plans on a policy with two producers after the statement was billed and partially paid.

Workaround: Guidewire is aware of this issue.

[Some catch-up invoice functionality was lost when InvoiceMatcher was replaced with InvoiceAssembler \(CC-56877\)](#)

Issue: The catch-up invoice feature lost some functionality when **InvoiceMatcher** was replaced with **InvoiceAssembler**. For example, the ability to create a catch-up invoice with a direct bill change.

Workaround: This is a new feature that is not fully implemented. Guidewire is aware of this issue.

[BillingCenter did not process policy changes done after the policy expiration date \(CC-56852\)](#)

Issue: During BillingCenter - Policy Center integration, BillingCenter does not process policy changes made after the policy expiration date. An illegal state exception displays on both the BillingCenter and PolicyCenter consoles.

Workaround: Guidewire is aware of this issue.

[Using Edit Next Invoice Date may not change the invoice day of month for future invoices \(CC-56840\)](#)

Issue: When using the **Edit Next Invoice Date** on the **Account → Invoices** screen to change the **invoice day of month**, BillingCenter continues to use the original **invoice day of month**. This means that the account's future monthly invoices will still be created with the old **invoice day of month**. This could also affect held charges if their items get carried forward because BillingCenter will not see the modified invoices that have a new **invoice day of month**.

Workaround: Guidewire is aware of this issue.

[After reassigning an item to a new payer, the old account reflected incorrect information \(CC-56683\)](#)

Issue: After an item was reassigned to a new agency bill owner, the old account reflected the old information and still tracked the reassigned item as a past due amount.

Workaround: Use **one** of the following workarounds to have the account summary page reflect the correct information:

- From the Cycles screen, advance the date past the due date of the 4th statement and then run the Statement Billed and Statement Due batch processes. The statement status changes to Closed.
- From the Agency Bill plan, set the On item moved and On payment schedule change options of the Affect Billed Statement section to Yes.

[Moving a Billed invoice item should negate the direct bill commission paid, not the agency bill commission paid \(CC-56656\)](#)

Issue: When an item is moved from a **Billed** invoice to a **Planned** invoice, an earnings transaction is created for the Payable amount on the commission. However, if the commission was earned and paid, it is not being negated.

Workaround: Guidewire is aware of this issue.

Policy period transfer from a Null producer does not transfer any commission to the new producer (CC-56554)

Issue: When doing a retroactive transfer of a policy from a producer not earning commission to a new primary producer, no commission reserve is created for the new producer. If there is no original commission, BillingCenter sets the new producer commission to zero and ignores any commission rate associated with the new producer.

Workaround: Guidewire is aware of this issue.

After a reversal, the distributed payment is still being reserved for the premium payment (CC-56538)

Issue: The disbursement amount is not fully moved to disbursement and does not show up on the **Disbursement** page although an entire payment has been reversed and the associated charge had been assigned to another account.

Workaround: This is a new feature that is not fully implemented. Guidewire is aware of this issue.

Recalculating the distribution pays all negative items including those that are Unbilled (CC-56498)

Issue: When overriding or recalculating the distribution on either the **New Payment Wizard** or **New Direct Bill Payment** screen, BillingCenter runs the pro-rata distribution strategy and pays any negative items that appear on the distribution. This actions pays all negative items including **Unbilled** items.

Workaround: To prevent the distribution of all negative items, only display the items you want to include in the distribution on the payment screens or set the **Distribution Method** on the **Distribution** screen to **Up to Next Invoice**.

The applied net amount on an agency bill statement was not accurately calculated after reassignment of a partially paid item (CC-56153)

Issue: The payment applied net amount on an agency bill statement was not correctly calculated after a partially paid agency bill item was reassigned to another producer.

Workaround: Guidewire is aware of this issue.

Errors occur in the totals for the producer's rate on the Transactions screen (CC-55835)

Issue: A difference occurs in the totals shown on the **Transactions** screen for a producer when selecting **Reserves** and then choosing either the **Last 30 Days** or **Show All** filter.

Workaround: Guidewire is aware of this issue.

A duplication problem occurred with the PA SMessageTransport and BillingMessage Transport (CC-55665)

Issue: The PASMessage and Billing Message transports use a unique string (transaction ID) to avoid duplicate processing. They store the string in the message payload. However, if the change to the message payload is not committed before the message is sent, a failure can occur with the sending system before the receiving system has completed processing that can result in a duplicate message being created with a different transaction ID.

Workaround: Commit the message payload before sending PolicyCenter-BillingCenter integration messages so that the payload will not be lost in the event that PolicyCenter is down when BillingCenter processes the message. This can be done by committing message payload in **BillingMessageRequest** plugin.

New collateral charges and invoice amounts are being generated based on previously paid collateral amounts, not the previously billed or planned amounts (CC-55209)

Issue: New collateral charges and invoice amounts are being generated based on the amount of collateral previously paid and do not account for amounts of collateral previously Billed or Planned (not yet billed). This results in an over billing of the collateral amount on the invoices.

Workaround: Guidewire is aware of this issue.

During the charge invoicing process, the producer statements for the month of February were not being created (CC-54952)

Issue: When creating a new agency bill policy with a monthly payment plan, ten installments, and a down payment the February statement was not generated. Instead, the February item was placed on the invoice for the next month.

Workaround: This is part of a new feature that is not fully implemented. Guidewire is aware of this issue.

No commission reserve and onset transfer transaction was created when a direct bill item was reassigned to an agency bill producer (CC-54923)

Issue: When a direct bill policy has no specified primary producer and one of the policy's item is reassigned to an agency bill producer, no commission reserve and onset transfer transaction created. This was because there is no initial reserve.

Workaround: Guidewire is aware of this issue.

Need to be able to reverse remainder items when only redistributing commission on a subset of invoice items (CC-53069)

Issue: When redistributing commission on two items with modified gross amounts the original remainder item is reversed. A new remainder item is created that is specific to the calculated remainder of the two items.

The problem is that the original remainder item was calculated across all invoice items, and the new remainder item was calculated across only those two invoice items. The math for those two calculations is not equivalent in any case where there is a remainder.

Workaround: Guidewire is aware of this issue.

The IBillingCenterAPI does not handle account distribution for policy level billing options correctly (CC-52710)

Issue: When a payment is directed towards a particular policy, there are two billing options available from for use Account → Distribution → Policy-Level Billing:

- Ignore and distribute as if account-level
- Release extra funds to account

However, these options are not working correctly.

Workaround: Guidewire is aware of this issue.

Billing instructions for Issuance and Policy Change must work with premium and other charges (Premium Reporting Only) (CC-52032)

Issue: This issue is for Premium Reporting policy periods only. Users are note allowed to enter premium charges as part of an issuance or policy change and process them as any other policy period. This issue applies to all billing instructions types.

Workaround: Guidewire is aware of this issue.

[Automatic distribution to policies not working correctly \(CC-51511\)](#)

Issue: Automatic distribution to policies on **Account → Disbursement** screen does not distribute the payment correctly.

Workaround: Guidewire is aware of this issue.

[Policy Period Unapplied should not be used to hold funds \(CC-57487\)](#)

Issue: The `PolicyPeriod.Unapplied` T-account or `PolicyPeriod.payChargesUsingAccountUnappliedFunds()` method is deprecated. So, it should not be used to pay invoice items. Because policy periods can have invoice items that belong to more than one payer, invoice items will now be paid by a payer's funds. `PolicyPeriod.Unapplied` will be removed in a future release.

Workaround: Any money that ends up in a `PolicyPeriod.Unapplied` T-account but should logically be located elsewhere, such as a payer or producer account.

[Problems occurred with the use of negative collateral charges \(CC-51312\)](#)

Issue: When issuing a negative collateral charge, the amount gets subtracted from a `Charge Due` T-account and a payer's `Unapplied` T-account. This causes an imbalance between `collateral reserve` and `collateral held` when issuing a negative charge.

Workaround: Guidewire is aware of this issue.



chapter 36

Guidewire BillingCenter 3.0.0 Release Notes

Release 3.0.0.52

Copyright © 2001-2009 Guidewire Software, Inc. All rights reserved. Guidewire, Guidewire Software, Guidewire ClaimCenter, Guidewire PolicyCenter, and the Guidewire logo are registered trademarks, and Guidewire Insurance Suite, Guidewire BillingCenter, Guidewire ContactCenter, and Guidewire Studio are trademarks of Guidewire Software, Inc. All other product names are the trademarks of their respective companies.

This product includes information that is proprietary to Insurance Services Office, Inc (ISO). Where ISO participation is a prerequisite for use of the ISO product, use of the ISO product is limited to those jurisdictions and for those lines of insurance and services for which such customer is licensed by ISO.

This documentation is published as Guidewire Confidential. The contents of this documentation, including product architecture details and APIs, are considered confidential and are fully protected by customer licensing confidentiality agreements and signed Non-Disclosure Agreements (NDAs).

This product includes software developed by the Apache Software Foundation (<http://www.apache.org>).

Overview

These release notes contain the following sections:

- Release Information
- Installing This Release
- Changes in this Release
- Known Issues and Limitations

Release Information

These release notes apply only to this release of Guidewire BillingCenter.

Version Number

This release of Guidewire BillingCenter is 3.0.0.52.

Support

This document includes important information about the current release of Guidewire BillingCenter. For assistance with this software release, contact Guidewire Customer Support. Contact support on the Web at <http://portal.guidewire.com>, by email at support@guidewire.com, or by phone at +1-650-356-4955.

Installing This Release

For installation information, see the *BillingCenter Installation Guide*.

Changes in this Release

This section describes the product changes in this release.

- Configuration and GScript API Changes
- Base PCF File Changes

Configuration and GScript API Changes

For a list of configuration and GScript API changes between BillingCenter 2.1.5 and BillingCenter 3.0.0, [click here](#) (requires the `readme_files` directory on your local disk). This is a Microsoft Excel file that details changes in the following:

- entity model
- GScript API
- typelists
- display keys

Base PCF File Changes

All links below require the `readme_files` directory on your local disk.

BillingCenter release 2.1.5 to 3.0.0

- To view a report of the changes in the base PCF files in the `modules/bc` directory, [click here](#).
- To view a report of the changes in the base PCF files in the `modules/platform` directory, [click here](#).

Known Issues and Limitations

Note: For maintenance releases, it is Guidewire policy to defer fixing configuration issues if they require merging files during the upgrade. Workarounds to many of these issues are listed in the following sections. The goal of this policy is to make upgrades as straightforward as possible.

This section describes known issues with this release of Guidewire BillingCenter.

WebSphere timeout during initial startup (CC-4904)

Issue: The initial creation of the BillingCenter database may take longer than the default WebSphere startup timeouts will allow, and WebSphere may abort the startup before the database upgrade has completed. In this situation, the `Default_Server_stderr.log` file will show an interruption in the “Applying upgrade step ### of #####” log messages.

Workaround: Increase the Ping Timeout and Ping Initial Timeout values for WebSphere. These can be found in the WSADMIN Console; click **Nodes**, drill down to your application server, click the **Advanced** tab, and increase the values.

Acrobat Sample document template does not allow extra fields (PL-1416)

Issue: Currently, the example Acrobat document template uses two fields, and this requires the same number of fields in `SampleAcrobat.pdf.descriptor` and `SampleAcrobat.pdf` files.

Workaround: To allow more fields in the template and documents created from it, you must update its descriptor file.

Incorrect additivity statements in the default logging.properties file (CC-50615)

Issue: Some additivity statements in the `logging.properties` file are incorrect and therefore, ineffective. An additivity statement attempts to keep the category from sending its log messages up to its parent, preventing duplicate log messages across files.

Workaround: In the `logging.properties` file (located at `modules/cc/config/logging`), change:

```
log4j.additivity.PluginsLog=false  
to  
log4j.additivity.Plugin=false
```

BillingCenter login remains active in new browser window (CC-4152)

Issue: When you log into BillingCenter in a browser window, then you close that window and open a new one, in some instances you can access BillingCenter pages without needing to log in again.

This is an issue with some versions of Microsoft Internet Explorer. When multiple browser windows are running in the same process, they can share session information, and this allows them to share a single login with BillingCenter. Early versions of Internet Explorer provided a **Launch browser windows in a separate process** option, which, if enabled, would avoid this situation. Some versions of Internet Explorer do not allow you to set this option, and enable or disable it automatically. You can review the relevant Microsoft Knowledge Base Article 240928. However, you may experience a situation in which Internet Explorer does not correctly enable this option, especially if you manually disabled it in an earlier version and then later upgraded. In this case, the multiple browser windows are run in the same process, exhibiting the undesired behavior.

Workaround: The simplest solution is to always click **Logout** at the top of the BillingCenter window before closing your browser.

If desired, you can also take steps to manually enable the Internet Explorer option. This requires you to edit the Windows registry on the affected computer.

WARNING Editing the registry is a risky procedure, and any improper changes may result in permanent data loss and require Microsoft Windows to be completely reinstalled on that computer. To make this change, create or edit the

`[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\BrowseNewProcess]` registry entry. Set the data type to `REG_SZ`, and then set the data value to `yes`.

[WebSphere timeout during initial startup \(CC-4904\)](#)

Issue: The initial creation of the BillingCenter database may take longer than the default WebSphere startup timeouts will allow, and WebSphere may abort the startup before the database upgrade has completed. In this situation, the `Default_Server_stderr.log` file shows an interruption in the `Applying upgrade step ### of ####` log messages.

Workaround: Increase the Ping Timeout and Ping Initial Timeout values for WebSphere. These can be found in the WSADMIN Console. Click **Nodes**, drill down to your application server, click the **Advanced** tab, and increase the values.

[BillingCenter displays many warnings when regenerating the toolkit for regenerating Javadoc \(CC-55564\)](#)

Issue: This issue occurs when running the `regen-toolkit` command.

Workaround: Guidewire is aware of this issue and will address it in a future release.

[On AIX without an X11 server, BillingCenter may report that an X11 resource is missing \(CC-5889\)](#)

Issue: BillingCenter does not require an X11 server; however other components of the BillingCenter environment (such as Java) may inadvertently have this requirement. If so, BillingCenter may report that an X11 resource is missing when it is starting up.

Workaround: Install an X11 server, or contact Guidewire support to determine if the dependency can be removed.

[Port conflict when starting BillingCenter with JMX RMI adapter enabled and Microsoft Outlook running \(CC-6964\)](#)

Issue: When Microsoft Outlook starts, it chooses several ports to use for its connection to an Exchange server. In some instances, Outlook may use port 1099, which is a port that the BillingCenter JMX RMI adapter can use, *if enabled*. The BillingCenter server then reports a port conflict during startup when it tries to use that port for the RMI adapter.

Workaround: Close Outlook to free the port, and then start the BillingCenter server again. You can also change the port that the RMI adapter uses by editing the `JMXManagementPlugin` through Guidewire Studio.

[Studio PCF Editor treats commented-out property values as errors \(PL-4582\)](#)

Issue: If you attempt to comment out a property value in the Studio PCF editor, Studio treats this as an error.

Workaround: None. Guidewire is aware of this issue and will address it in a future release.

[Incomplete implementation of non-English locale-specific database search and sort \(PL-8108\)](#)

Issue: Guidewire has begun implementation of database searching and sorting appropriate to non-English locales. This release of BillingCenter contains an incomplete implementation of this work. The changes will not affect customers implementing case-insensitive searches using standard English sort criteria.

As part of this work, the current release renames the `supportsCaseInsensitiveSearch` attribute on column to `supportsLinguisticSearch`.

Workaround: Guidewire is aware of this issue and will address it in a future release.

Server exception during startup (PL-8167)

Issue: During server startup, there can be a Tomcat server exception. However, after the exception, the server does start up. This is caused by the Tomcat application server attempting to either save or restore sessions using serialization. The session restore causes an exception because BillingCenter sessions contain objects which rely on the metadata being started up.

Workaround: None. The Tomcat application server does start up after the error.

Zone data with missing values cannot be loaded (PL-9123)

Issue: Zone data, which is used for associating postal codes with cities and regions, does not load if it is incomplete.

Workaround: Guidewire is aware of this issue and will address it in a future release.

H2 development database creates LONGTEXT fields as VARCHAR(65000), instead of as CLOB as in Oracle and SQL Server (PL-9314)

Issue: Guidewire creates schema fields defined as LONGTEXT as LONGVARCHAR columns, instead as CLOB as in the Oracle or SQL Server databases. In addition to the inconsistency involved, this mandates a field length of 65,000 characters or less in LONGTEXT columns in the H2 database.

Workaround: Define a LONGTEXT field as a VARCHAR(...) with the required size, for example, as VARCHAR(120000).

BillingCenter does not display empty report folders (PL-1281)

Issue: BillingCenter does not display a report folder in the Administration → Report Admin page if that report folder is empty.

Workaround: Within InetSoft Enterprise Manager, insert a replet into the empty report folder. You can then mark this replet as not visible (using the InetSoft Enterprise Manager), which causes the replet to not show in BillingCenter. However, BillingCenter does display the parent folder.

Changes to encryption column can cause encrypted columns to shrink (PL-9665)

Issue: If you change the encryption algorithm in such a way that the encrypted columns shrink, then it causes the resulting upgrade to crash on SQLServer.

Workaround: Guidewire is aware of this issue and will address it in a future release.

BillingCenter does not display empty report folders (PL-1281)

Issue: BillingCenter does not display a report folder in the Administration → Report Admin page if that report folder is empty.

Workaround: Within InetSoft Enterprise Manager, insert a replet into the empty report folder. You can then mark this replet as not visible (using the InetSoft Enterprise Manager), which causes the replet to not show in BillingCenter. However, BillingCenter does display the parent folder.

Changes to encryption column can cause encrypted columns to shrink (PL-9665)

Issue: If you change the encryption algorithm in such a way that the encrypted columns shrink, then it causes the resulting upgrade to crash on SQLServer.

Workaround: Guidewire is aware of this issue and will address it in a future release.

Incorrect totals in the policy payment schedule occurred on agency bill statements after a change was made to one of two producers on a policy (CC-56925)

Issue: The producer statement does not show the correct dollar amount after a change is made to one of the policy's payment plans on a policy with two producers after the statement was billed and partially paid.

Workaround: Guidewire is aware of this issue and will address it in a future release.

Some catch-up invoice functionality was lost when InvoiceMatcher was replaced with InvoiceAssembler (CC-56877)

Issue: The catch-up invoice feature lost some functionality when `InvoiceMatcher` was replaced with `InvoiceAssembler`. For example, the ability to create a catch-up invoice with a direct bill change.

Workaround: This is a new feature that is not fully implemented. Guidewire is aware of this issue and will address it in a future release.

BillingCenter did not process policy changes done after the policy expiration date (CC-56852)

Issue: During BillingCenter - Policy Center integration, BillingCenter does not process policy changes made after the policy expiration date. An illegal state exception displays on both the BillingCenter and PolicyCenter consoles.

Workaround: Guidewire is aware of this issue and will address it in a future release.

Using Edit Next Invoice Date may not change the invoice day of month for future invoices (CC-56840)

Issue: When using the `Edit Next Invoice Date` on the `Account → Invoices` screen to change the `invoice day of month`, BillingCenter continues to use the original `invoice day of month`. This means that the account's future monthly invoices will still be created with the old `invoice day of month`. This could also affect held charges if their items get carried forward because BillingCenter will not see the modified invoices that have a new `invoice day of month`.

Workaround: Guidewire is aware of this issue and will address it in a future release.

The `IDataExtractionAPI` is being deprecated as a default Webservice for security concerns. (CC-56710)

Issue: The `IDataExtractionAPI` is being deprecated as a default Webservice for security concerns. Exposing this method allows arbitrary Gosu to be run on a production server.

Workaround: For customers that are currently using the `IDataExtractionAPI` in integrations, Guidewire recommends creating a targeted WebService that takes only the parameters, and restricts access to the objects needed. Customer do have the ability to re-enable this WebService if they need to continue to use it, but due to its vulnerability to attacks, it is highly discouraged.

A duplicate set of invoices gets created if two policies are assigned at the same time (CC-56819)

Issue: Adding a second list bill policy creates a second set of invoices for a payer with the same dates as the existing invoices.

Workaround: Guidewire is aware of this issue and will address it in a future release.

Changing the billing method caused mismatched statement information (CC-56716)

Issue: Changing the billing method from direct bill to agency bill caused information on the statement to differ from information in the `Payment wizard`. This issue affects commission amounts and occurred when commission earnings in the producer's commission plan was set to `On Binding`.

Workaround: Guidewire is aware of this issue and will address it in a future release.

The closure status and earned premium amount incorrectly populated the new term of a policy period (CC-56691)

Issue: Incorrect closer status and earned premium amounts incorrectly populated the terms of a new policy period when the insured first closed the policy and then renewed it.

Workaround:

After reassigning an item to a new payer, the old account reflected incorrect information (CC-56683)

Issue: After an item was reassigned to a new agency bill owner, the old account reflected the old information and still tracked the reassigned item as a past due amount.

Workaround: Guidewire is aware of this issue and will address it in a future release.

Making changes to a payment plan associated with a policy caused the default payer to be ignored (CC-56678)

Issue: Changing the payer of a charge from the owner to another payer and then making a change to the payment plan caused the charge to land on the owners invoice instead of the payer's invoice.

Workaround: Guidewire is aware of this issue and will address it in a future release.

Suspense items for direct bill payments are unavailable (CC-56667)

Issue: BillingCenter is deprecating Unapplied T-account Policy Period and the recommended process for sheltering funds will be to use direct bill suspense payments. However, direct bill suspense items are currently under construction and unavailable for this release.

Workaround: This is a new feature that is not fully implemented. Guidewire is aware of this issue.

Moving a Billed invoice item should negate the direct bill commission paid, not the agency bill commission paid (CC-56656)

Issue: When an item is moved form a Billed invoice to a Planned invoice, an earnings transaction is created for the Payable amount on the commission. However, if the commission was earned and paid, it is not being negated.

Workaround: Guidewire is aware of this issue and will address it in a future release.

Making changes to the payment plan may cause problems with the distribution (CC-56583)

Issue: Making changes to the payment plan may cause problems with the distribution when payment distribution is set to **Up to Amount Under Contract**. Instead of making the payment based on a call, BillingCenter waits for the batch process to pick up the unused payments or unapplied funds for distribution.

Workaround: This is a new feature that is not fully implemented. Guidewire is aware of this issue.

Policy period transfer from a Null producer does transfer any commission to the new producer (CC-56554)

Issue: When doing a retroactive transfer of a policy from a producer not earning commission to a new primary producer, no commission reserve is created for the new producer. If there is no original commission, BillingCenter sets the new producer commission to zero and ignores any commission rate associated with the new producer.

Workaround: Guidewire is aware of this issue and will address it in a future release.

After a reversal, the distributed payment is still being reserved for the premium payment (CC-56538)

Issue: The disbursement amount is not fully moved to disbursement and does not show up on the **Disbursement** page although an entire payment has been reversed and the associated charge had been assigned to another account.

Workaround: This is a new feature that is not fully implemented. Guidewire is aware of this issue.

Recalculating the distribution pays all negative items including those that are Unbilled (CC-56498)

Issue: When overriding or recalculating the distribution on either the **New Payment Wizard** or **New Direct Bill Payment** screen, BillingCenter runs the pro-rata distribution strategy and pays any negative items that appear on the distribution. This actions pays all negative items including **Unbilled** items.

Workaround: To prevent the distribution of all negative items, only display the items you want to include in the distribution on the payment screens or set the **Distribution Method** on the **Distribution** screen to **Up to Next Invoice**.

The applied net amount on an agency bill statement was not accurately calculated after reassignment of a partially paid item (CC-56513)

Issue: The payment applied net amount on an agency bill statement was not correctly calculated after a partially paid agency bill item was reassigned to another producer.

Workaround: Guidewire is aware of this issue and will address it in a future release.

Errors occurred in the totals for the producer's rate on the Transactions screen (CC-55835)

A difference occurred in the totals shown on the **Transactions** screen for a producer when selecting **Reserves** and then choosing either the **Last 30 Days** or **Show All** filter.

Workaround: Guidewire is aware of this issue and will address it in a future release.

A duplication problem occurred with the PA SMessageTransport and BillingMessage Transport (CC-55665)

Issue: The PASMessage and Billing Message transports use a unique string (transaction ID) to avoid duplicate processing. They store the string in the message payload. However, if the change to the message payload is not committed before the message is sent, a failure can occur with the sending system before the receiving system has completed processing that can result in a duplicate message being created with a different transaction ID.

Workaround: Commit the message payload before sending PolicyCenter-BillingCenter integration messages so that the payload will not be lost in the event that PolicyCenter is down when BillingCenter processes the message. This can be done by committing message payload in **BillingMessageRequest** plugin.

New collateral charges and invoice amounts are being generated based on previously paid collateral amounts, not the previously billed or planned amounts (CC-55209)

Issue: New collateral charges and invoice amounts are being generated based on the amount of collateral previously paid and do not account for amounts of collateral previously **Billed** or **Planned** (not yet billed). This results in an over billing of the collateral amount on the invoices.

Workaround: Guidewire is aware of this issue and will address it in a future release.

Point-in-Time transfer does not handle situations where sourcePolicyCommission and targetPolicyCommission is set to Null (CC-54969)

Issue: When a point-in-time transfer occurs and the `sourcePolicyCommission` is set to `Null`, there is no reserve transfer; when the `targetPolicyCommission` is set to `Null`, the commission reserve for all invoice items remains on the item commission for the source.

Workaround: Guidewire is aware of this issue and will address it in a future release.

During the charge invoicing process, the producer statements for the month of February were not being created (CC-54952)

Issue: When creating a new agency bill policy with a monthly payment plan, ten installments, and a down payment the February statement was not generated. Instead, the February item was placed on the invoice for the next month.

Workaround: This is part of a new feature that is not fully implemented. Guidewire is aware of this issue.

No commission reserve and onset transfer transaction was created when a direct bill item was reassigned to an agency bill producer (CC-54923)

Issue: When a direct bill policy has no specified primary producer and one of the policy's item is reassigned to an agency bill producer, no commission reserve and onset transfer transaction created. This was because there is no initial reserve.

Workaround: Guidewire is aware of this issue and will address it in a future release.

BillingCenter should store dates as datetime, not dateonly (CC-53850)

Issue: Billing Center should store policy period effective date, modification date, and expiration date as `datetime`, not `dateonly`. This change will make BillingCenter consistent with other Guidewire products and will help facilitate integration between the products.

Workaround: Guidewire is aware of this issue and will address it in a future release.

Paying a Billed negative item that is not due makes the policy period past due (CC-53398)

Issue: Paying a Billed negative item, that is not yet due and is either agency bill or direct bill makes the policy period Past Due. Commission cannot be earned for this invoice until the statement or invoice becomes Due.

Workaround: Guidewire is aware of this issue and will address it in a future release.

Need to be able to reverse remainder items when only redistributing commission on a subset of invoice items (CC-53069)

Issue: When redistributing commission on two items with modified gross amounts the original remainder item is reversed. A new remainder item is created that is specific to the calculated remainder of the two items.

The problem is that the original remainder item was calculated across all invoice items, and the new remainder item was calculated across only those two invoice items. The math for those two calculations is not equivalent in any case where there is a remainder.

Workaround: Guidewire is aware of this issue and will address it in a future release.

The IBillingCenterAPI does not handle account distribution for policy level billing options correctly (CC-52710)

Issue: When a payment is directed towards a particular policy, there are three billing options available from for use Account → Distribution → Policy-Level Billing:

- Keep extra funds with the policy
- Ignore and distribute as if account-level
- Release extra funds to account

However, these options are not working correctly.

Workaround: Guidewire is aware of this issue and will address it in a future release.

Billing instructions for Issuance and Policy Change must work with premium and other charges (CC-52032)

Issue: Need to provide the ability to enter premium charges as part of an issuance or policy change and process them as any other policy period. This issue applies to all billing instructions types.

Workaround: Guidewire is aware of this issue and will address it in a future release.

Automatic disbursement to policies not working correctly (CC-51511)

Issue: Automatic disbursement to policies on Account → Disbursement screen does not disburse the payment correctly.

Workaround: Guidewire is aware of this issue and will address it in a future release.

Policy Period Unapplied should not be used to hold funds (CC-51316)

Issue: The Policy Period Unapplied T-account is deprecated. So, it should not be used to pay invoice items. Because policy periods can have invoice items that belong to more than one payer, invoice items will now be paid by a payer's funds. Any money that ends up in a Policy Period Unapplied T-account should logically be located elsewhere, such as a payer or producer account.

Workaround: Guidewire is aware of this issue and will address it in a future release.

Problems occurred with the use of negative collateral charges (CC-51312)

Issue: When issuing a negative collateral charge, the amount gets subtracted from a Charge Due T-account and a payer's Unapplied T-account. This causes an imbalance between collateral reserve and collateral held when issuing a negative charge.

Workaround: Guidewire is aware of this issue and will address it in a future release.

Held account-level charges are not showing up on the Desktop (CC-44581)

Issue: Account-level charges, such as a reversal fee, are not showing up in the user interface on Desktop → Held Charges → Held Charges screen. These charges should be available.

Workaround: Guidewire is aware of this issue and will address it in a future release.