



# Guidewire ClaimCenter®

## New and Changed Guide

RELEASE 9.0.0

Copyright © 2001-2016 Guidewire Software, Inc.

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, Guidewire Spotlight, Gosu, Adapt and succeed, 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 ClaimCenter

Product Release: 9.0.0

Document Name: ClaimCenter New and Changed Guide

Document Revision: 23-June-2016

# Contents

<b>About ClaimCenter Documentation .....</b>	<b>13</b>
Conventions in This Document .....	14
Support .....	14

## Part I New and Changed in 9.0.0

<b>1 New and Changed in the Application in 9.0.0.....</b>	<b>17</b>
New in the Application in 9.0.0 .....	17
Business Rules for Activities .....	17
Straight-through Invoice Processing (STIP).....	18
Viewing Documents Associated with Vendor Contacts.....	18
Changed in the Application in 9.0.0 .....	18
Document Management Changes .....	18
Subrogation Improvements.....	19
Exposure Constraint Configuration .....	19
Address field on Check .....	19
Granular Recovery Management .....	19
Authority Limits Configuration .....	19
Financial Summary Screen Configuration .....	19
Login Screen No Longer Saves User Information .....	20
<b>2 New and Changed in Configuration in 9.0.0.....</b>	<b>21</b>
Changes to Configuration Files .....	21
Changes to Configuration APIs .....	22
Changes to Exceptions in ClaimCenter .....	22
Changes to the Data Model.....	23
Changes to Data Model Entity Definitions .....	23
Entity Types Can Declare Required Properties .....	23
Typecode Identifier Customization.....	24
Delegate Attribute requiresType .....	24
Changes to Guidewire Studio .....	24
Studio Editors Provide Editable XML .....	24
Running Guidewire Studio .....	25
Changes to Configuration Parameters .....	25
Removed Configuration Parameters .....	25
Added Configuration Parameters .....	25
Changed Configuration Parameters .....	26
Changes Related to PCF Files .....	26
Use Type-specific Input and Cell PCF Elements Instead of Generic Ones .....	26
PCF Symbols are Case-sensitive.....	27
Other PCF Changes.....	27
Number and DateTime Types Have Been Replaced.....	27
PCF Changes Due to Upgrade .....	27
Changes Related to Display Keys.....	27
Display Keys are Case-sensitive .....	27
Display Key Folder and File Structure is Flattened .....	27
Access Display Keys with DisplayKey.get.....	27

<b>3</b>	<b>Changed in Globalization in 9.0.0 .....</b>	<b>29</b>
	Localizations Folder in Studio Has No Subfolders.....	29
<b>4</b>	<b>New and Changed in Installation and Upgrade in 9.0.0.....</b>	<b>31</b>
	Changed in Installation in 9.0.0 .....	31
	Apache Ant Installation Requirement Removed.....	31
	New in Upgrade in 9.0.0 .....	31
	Configuration Upgrade Tools .....	32
	Rolling Configuration Upgrade of ClaimCenter Cluster Members .....	32
	Custom Application Version Labels .....	32
	Upgrade-related System Tools Command Options.....	33
	Upgrade Flag Required for Upgrade.....	33
	Changed in Upgrade in 9.0.0 .....	33
	Changes to the Database Configuration File.....	33
<b>5</b>	<b>New and Changed in System Administration in 9.0.0.....</b>	<b>35</b>
	New in System Administration in 9.0.0 .....	35
	Batch Process Load Balancing .....	35
	Messaging and Startable Server Load Balancing .....	36
	System Property Pass Through Option.....	36
	Graceful Server Shutdown .....	36
	New in Guidewire Server Tools .....	36
	New Address Deletion Work Queue.....	36
	Changed in System Administration in 9.0.....	37
	Usage Tools Command Removed.....	37
	Administration Command Prompt Tools .....	37
	Changes to Logging Infrastructure .....	37
	Modifications to the Server Registry Element in config.xml.....	38
	Removal of Support for JGroups Clustering.....	38
	Removal of Dedicated Batch Server Requirement.....	38
	Setting Server System Properties .....	39
	Changes in Guidewire Server Tools .....	39
<b>6</b>	<b>New and Changed in Gosu in 9.0.0 .....</b>	<b>41</b>
	New in Gosu in 9.0.0 .....	41
	Structural Types .....	41
	Dynamic Types and Expando Objects .....	42
	Checked Arithmetic .....	42
	JSON Support .....	42

Changed in Gosu in 9.0.0 .....	43
Entity and Typelist Case Sensitivity Changes .....	43
Entity and Typelist Type System Changes .....	44
Entity Instance Static Literal Syntax Removed .....	45
Property Getter Methods Required for New Script Parameters .....	46
Normalization of Gosu Code .....	46
Metatype Properties and Methods No Longer Directly on the Type .....	46
Number, DateTime, Bean, List Types Removed .....	46
Changes to List Instantiation .....	46
Find Expressions Removed .....	47
Java Style Constructors Removed .....	47
Implicit Coercions Removed .....	47
XML API Changes .....	49
GX Model Changes .....	51
Gosu Property File Support Removed .....	51
Any uses Lines Must Be At the Top of the File .....	51
The typeis Operator Compile Time Changes .....	51
Annotations Changes .....	52
<b>7 New and Changed in Integration in 9.0.0 .....</b>	<b>53</b>
New in Integration in 9.0.0 .....	53
SystemToolsAPI Web Service Has New Cluster Management Methods .....	53
Changed in Integration in 9.0.0 .....	54
ISO Integration Changes .....	54
Changes to the ServiceRequestAPI Web Service .....	54
MailToAddress Renamed in Financials Web Services DTOs .....	55
RPCE Web Services Fully Removed .....	55
WS-I Web Service Changes to regen-soap-api and @WsiGenInToolkit .....	55
Cluster Methods in MaintenanceToolsAPI Moved to SystemToolsAPI .....	55
Document Management Changes .....	55
Public ID Length Is 64 Characters .....	58
Java Changes in 9.0.0 .....	58
<b>Part II</b>	
<b>New and Changed in 8.0 Maintenance Releases</b>	
<b>8 New and Changed in ClaimCenter 8.0.5 .....</b>	<b>65</b>
Secure Transport (HTTPS) for Free-Text Search .....	65
WSS4J and xmlsec Library Changes .....	65
<b>9 New and Changed in ClaimCenter 8.0.4 .....</b>	<b>67</b>
Changed in ClaimCenter 8.0.4 .....	67
Template Tool Modified to Manage Email and Note Templates .....	67
Bulk Purge Batch Processing .....	67
<b>10 New and Changed in ClaimCenter 8.0.3 .....</b>	<b>69</b>
New in ClaimCenter 8.0.3 .....	69
Changed in ClaimCenter 8.0.3 .....	69
Support for Oracle Partitioned Indexes .....	69
Improving Claim Search Performance with Oracle Databases .....	70
Change to Configuring ClaimCenter Exit Point to ContactManager .....	70
Upgrade to Free-text Search .....	70

<b>11 New and Changed in ClaimCenter 8.0.2 . . . . .</b>	<b>73</b>
New in ClaimCenter 8.0.2 . . . . .	73
New Method to Import a Contact and Save It, Plus Two New Exceptions . . . . .	73
New Method to Save a Related ClaimCenter Contact in ContactManager . . . . .	74
New Plugin Interface InboundIntegrationMessageReply . . . . .	74
Data Change Menu Link . . . . .	74
Changed in ClaimCenter 8.0.2 . . . . .	74
Removed Configuration Parameters . . . . .	74
Integration Changes . . . . .	74
<b>12 New and Changed in ClaimCenter 8.0.1 . . . . .</b>	<b>77</b>
New in ClaimCenter 8.0.1 . . . . .	77
Search for Claims by Claim Contact . . . . .	77
New Support for OSGi Plugins . . . . .	78
Gosu Using Clause Now Supports Finally Clause . . . . .	78
Changed in ClaimCenter 8.0.1 . . . . .	78
Changes to Services . . . . .	78
Changes to Archiving . . . . .	78
Aggregate Limit Enhancements . . . . .	79
Conversion of Web Services from RPCE to WS-I . . . . .	79
Changes to Web Service Transaction IDs . . . . .	80
Change to Inbound Integration Configuration . . . . .	80
Gosu Suppress Warnings Annotation . . . . .	80
Gosu Compound Assignment Operators for Logical AND and Logical OR . . . . .	80

## Part III

### New and Changed in 8.0.0

<b>13 New and Changed in the Application in 8.0.0 . . . . .</b>	<b>85</b>
New in the Application in 8.0.0 . . . . .	85
Services . . . . .	85
Changing the Screen Layout . . . . .	86
Changed in the Application in 8.0.0 . . . . .	86
Special Handling – Customer Service Tiers . . . . .	86
Weighted Workload . . . . .	86
Multicurrency Reserving . . . . .	87
Aggregate Limits . . . . .	87
Changed Claim Exception Batch Processing . . . . .	87
Changes to Document Assistant . . . . .	88
Changes to the Address Book Tab . . . . .	88
Contact Changes No Longer Require Permissions . . . . .	88
Changes to Integration with ContactManager . . . . .	88
<b>14 New and Changed in Configuration in 8.0.0 . . . . .</b>	<b>89</b>
Important Changes to the Configuration Module . . . . .	89
Changes to the Data Model . . . . .	90
Changes to the Generate Dictionary Command . . . . .	90
Changes to Guidewire Studio . . . . .	90
Studio Now Built on IntelliJ IDEA Community Edition . . . . .	90
Gosu Tester . . . . .	90
Display Keys Editor . . . . .	90
Studio Features That Are No Longer Available . . . . .	90

Changes to Configuration Parameters .....	91
Removed Configuration Parameters .....	91
Added Configuration Parameters .....	91
Changed Configuration Parameters .....	92
Changes to Web Services .....	92
Changes Related to PCF Files .....	92
Template Page PCF Files Removed .....	92
Miscellaneous PCF Changes .....	92
Changes Related to Integrating with ContactManager .....	94
Removal of Preload.txt Lines that Include Run Level .....	95
<b>15 New and Changed in Globalization in 8.0.0 .....</b>	<b>97</b>
New in Globalization in 8.0.0 .....	97
Separation of Language and Regional Format .....	98
Base Configuration Language Support .....	98
Base Configuration Region Support .....	98
Base Configuration Currency Support .....	99
Base Configuration Geographical Support .....	99
Phone Number Data and Phone Validator Gosu Class .....	100
Automatic Address Completion and Fill-in Plugin Functionality .....	100
Ability to Provide a Sort Order for the Typecodes in a Typelist .....	101
Ability to Localize Script Parameter Descriptions .....	101
Language Module Installation Utility .....	101
Globalization-related Configuration Parameters .....	102
Globalization-related Typelists .....	102
Ability to Set First Day of Week by Region .....	102
New in the ClaimCenter 8.0 Data Model .....	102
Changes in Globalization in 8.0.0 .....	103
Changes to Configuration Parameter MulticurrencyDisplayMode .....	104
Changes to Address Configuration 8.0 .....	104
Changes to Admin Data Localization .....	104
Changes to 8.0 Typelists .....	105
Changes to Files Used with Globalization .....	106
List Sort Methods Support Optional Locale Sensitive Sorting .....	107
<b>16 New and Changed in System Administration in 8.0.0 .....</b>	<b>109</b>
New in System Administration in 8.0.0 .....	109
Phone Number Normalizer Work Queue .....	109
User Workload Update Work Queue .....	109
Changed in System Administration in 8.0.0 .....	110
Changes to Database Configuration .....	110
Changes to Work Queue Configuration .....	112
Changes to Scheduled Purge Batch Processing .....	112
Changes to Database Statistics Updating .....	112
<b>17 New and Changed in Gosu in 8.0.0 .....</b>	<b>115</b>
New in Gosu in 8.0.0 .....	115
New Assert Statement .....	115
In For Loop Declarations, Local Variable is Now Optional .....	115
Final Variable Initialization Separate from Declaration .....	116
The new Operator is Now Optionally a Statement .....	117
Support Annotations on Function Parameters .....	117
Named Arguments in Annotations .....	117
New Support for Code Coverage Tools for Gosu Code .....	117
The Entity Touch API is Changed and No Longer Deprecated .....	117

Changed in Gosu 8.0.0 .....	118
Gosu is Now Case Sensitive In Most Cases .....	118
Bundle Changes from Gosu and Java .....	118
Gosu Map Enhancement Method Changes .....	118
Entity Literal Syntax is Deprecated .....	119
Entity Methods loadByKey, loadByPublicID, and remove Deprecated .....	119
Packages Changed to Reflect Public and Internal Status .....	119
Gosu Concurrency API Changes .....	121
The <> Inequality Operator Is Now Invalid .....	121
The Gosu Command Line Tool Built-in Editor Removed .....	121
Change in Gosu Named Parameters Usage .....	121
<b>18 New and Changed in Integration in 8.0.0 .....</b>	<b>123</b>
New in Integration in 8.0.0 .....	123
New Phone Number Normalizer Plugin .....	123
New MTOM Support for Results of Published WS-I Web Service .....	124
Multi-threaded Inbound Integration .....	124
New Messaging Destination Option for Messages Without Primary Object .....	124
Regenerating WSDL and XSD in Web Service Collections (regen-from-wsc) .....	125
New Plugin Interface IAggregateLimitTransactionPlugin .....	125
New Plugin Interface PolicyLocationSearchPlugin .....	125
Changed in Integration 8.0.0 .....	125
RPCE Web Services Deprecated .....	126
WS-I Web Services That Replace Deprecated RPCE Web Services .....	126
Existing WS-I Web Services Changes .....	130
Java Code Changes .....	131
Plugin Registry Changes in Studio .....	137
Document Management Integration Changes .....	137
ClaimCenter Contact Plugins Changes .....	139
PolicyCenter Product Model Import into ClaimCenter Changes .....	139

## Part IV

### Release Notes Archive

<b>19 Guidewire ClaimCenter 8.0.4 Release Notes .....</b>	<b>143</b>
Overview .....	144
Release Information .....	144
Version Number .....	144
Installing This Release .....	144
Upgrade Information .....	144
Support .....	145
Major Issues and Changes .....	145
Base PCF File Changes .....	145
Base Rule Changes .....	145
Changes in This Release Provided in Upgrade Diff Report .....	145
Improvements and Resolved Issues .....	145
Application Improvements and Resolved Issues .....	145
Platform Improvements and Resolved Issues .....	147
Known Issues and Limitations .....	149
ClaimCenter Known Issues .....	149
Studio/Platform Known Issues .....	151
<b>20 Guidewire ClaimCenter 8.0.3 Release Notes .....</b>	<b>153</b>
Overview .....	154

Release Information .....	154
Version Number .....	154
Installing This Release .....	154
Upgrade Information.....	154
Support .....	155
Major Issues and Changes .....	155
Base PCF File Changes.....	156
Base Rule Changes .....	156
Changes in This Release Provided in Upgrade Diff Report.....	156
Improvements and Resolved Issues .....	156
Platform Improvements and Resolved Issues.....	158
Documentation Improvements and Resolved Issues.....	163
Known Issues and Limitations .....	164
ClaimCenter Known Issues.....	164
Platform/Studio Known Issues .....	166
<b>21 Guidewire ClaimCenter 8.0.2 Release Notes .....</b>	<b>169</b>
Overview.....	170
Release Information .....	170
Version Number .....	170
Installing This Release .....	170
Upgrade Information.....	170
Support .....	171
Major Issues and Changes .....	171
Base PCF File Changes.....	172
Base Rule Changes .....	172
Changes in This Release Provided in Upgrade Diff Report.....	172
Improvements and Resolved Issues .....	172
Platform Improvements and Resolved Issues.....	174
Known Issues and Limitations .....	179
ClaimCenter Known Issues.....	179
Platform/Studio Known Issues .....	180
<b>22 Guidewire ClaimCenter 8.0.1 Release Notes .....</b>	<b>185</b>
Overview.....	185
Release Information .....	186
Version Number .....	186
Installing This Release .....	186
Support .....	186
Issues and Major Changes .....	186
Base PCF File Changes.....	186
Base Rule Changes .....	187
Changes in This Release Provided in Upgrade Diff Report.....	187
Improvements and General Issues .....	187
Known Issues and Limitations .....	196
ClaimCenter Known Issues.....	197
Platform/Studio Known Issues .....	198
<b>23 Guidewire ClaimCenter 8.0.0 Release Notes .....</b>	<b>201</b>
Overview.....	201
Release Information .....	202
Version Number .....	202
Installing This Release .....	202
Support .....	202

Issues and Major Changes .....	202
Base PCF File Changes.....	202
Base Rule Changes .....	203
Changes in This Release Provided in Upgrade Diff Report.....	203
Known Issues and Limitations .....	203
ClaimCenter Known Issues.....	203
Platform/Studio Known Issues .....	204
<b>24 Guidewire ClaimCenter 7.0.6 Release Notes .....</b>	<b>209</b>
Overview.....	209
Release Information .....	210
Version Number .....	210
Installing This Release .....	210
Support .....	210
Issues and Major Changes .....	210
Base PCF File Changes.....	210
Rules Changes.....	210
Changes in This Release Provided in Upgrade Diff Report.....	211
Improvements and General Issues .....	212
Known Issues and Limitations .....	213
ClaimCenter Known Issues.....	213
Studio/Platform Issues .....	214
<b>25 Guidewire ClaimCenter 7.0.5 Release Notes .....</b>	<b>217</b>
Overview.....	217
Release Information .....	218
Version Number .....	218
Installing This Release .....	218
Support .....	218
Issues and Major Changes .....	218
Base PCF File Changes.....	218
Rules Changes.....	218
Changes in This Release Provided in Upgrade Diff Report.....	219
Improvements and General Issues .....	220
Known Issues and Limitations .....	222
ClaimCenter Known Issues.....	223
Studio/Platform Issues .....	224
<b>26 Guidewire ClaimCenter 7.0.4 Release Notes .....</b>	<b>227</b>
Overview.....	228
Release Information .....	228
Version Number .....	228
Installing This Release .....	228
Support .....	228
Issues and Major Changes .....	228
Base PCF File Changes.....	228
Rules Changes.....	229
Changes in This Release Provided in Upgrade Diff Report.....	229
Improvements and General Issues .....	230
Known Issues and Limitations .....	233
ClaimCenter Known Issues.....	234
Studio/Platform Issues .....	234
<b>27 Guidewire ClaimCenter 7.0.3 Release Notes .....</b>	<b>237</b>
Overview.....	237

Release Information .....	238
Version Number .....	238
Installing This Release .....	238
Support .....	238
Issues and Major Changes .....	238
Base PCF File Changes.....	238
Rules Changes.....	238
Changes in This Release Provided in Upgrade Diff Report.....	239
Improvements and General Issues .....	240
Known Issues and Limitations .....	243
ClaimCenter Known Issues.....	244
Studio/Platform Issues .....	245
<b>28 Guidewire ClaimCenter 7.0.2 Release Notes .....</b>	<b>247</b>
Overview.....	248
Release Information .....	248
Version Number .....	248
Installing This Release .....	248
Support .....	248
Issues and Major Changes .....	248
Manual Checks Are Not Allowed to Be Negative (CLM-18333) .....	248
Base PCF File Changes.....	249
Rules Changes.....	249
Changes in This Release Provided in Upgrade Diff Report.....	249
Improvements and General Issues .....	250
Known Issues and Limitations .....	259
ClaimCenter Known Issues.....	260
Studio/Platform Issues .....	260
<b>29 Guidewire ClaimCenter 7.0.1 Release Notes .....</b>	<b>265</b>
Overview.....	265
Release Information .....	266
Version Number .....	266
Installing This Release .....	266
Support .....	266
Issues and Major Changes .....	266
ContactManager Integration .....	267
Geocoding Using Bing and MapPoint (PL-16708). ....	267
Changes to Archive Functionality.....	267
Catastrophe Heat Map Configuration.....	268
Guidewire Standard Reporting .....	268
Run ContactAutoSync and Clear Failed Work Items (CLM-17855) .....	270
ClaimCenter unable to delete a contact from Address Book (CLM-17300) .....	270
Integrating ClaimCenter 7.0.1 with PolicyCenter 7.0.0 or 7.0.1 (CLM-17669). ....	270
Speed up claim search by denormalizing Contact fields (CLM-17885). ....	271
Audit of Gosu classes (CLM-17516) .....	271
Upgrade iso.xsd to version 5.3 (CLM-17681) .....	272
ContactAutoSyncWorkItem update (CLM-17990). ....	272
Base PCF File Changes.....	273
Rules Changes.....	273
Changes in This Release Provided in Upgrade Diff Report.....	273
Improvements and General Issues .....	274
Known Issues and Limitations .....	286
ClaimCenter Known Issues.....	286
Studio/Platform Issues .....	287

---

<b>30 Guidewire ClaimCenter 7.0.0 Release Notes .....</b>	<b>291</b>
Overview .....	291
Release Information .....	292
Version Number .....	292
Support .....	292
Installing This Release .....	292
Upgrading Coverage Terms (CLM-15711, CLM-15906) .....	292
Changes in This Release Provided in Upgrade Diff Report .....	292
Known Issues and Limitations .....	293
ClaimCenter Known Issues.....	293
Platform/Studio Known Issues .....	296

# About ClaimCenter Documentation

The following table lists the documents in ClaimCenter 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 ClaimCenter 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 ClaimCenter.
<i>Upgrade Guide</i>	Describes the overall ClaimCenter upgrade process, and describes how to upgrade your ClaimCenter database from a previous major version. The intended readers are system administrators and implementation engineers who must merge base application changes into existing ClaimCenter application extensions and integrations.
<i>Configuration Upgrade Guide</i>	Describes how to upgrade your ClaimCenter configuration from a previous major version. The intended readers are system administrators and implementation engineers who must merge base application changes into existing ClaimCenter application extensions and integrations. The <i>Configuration Upgrade Guide</i> is published with the Upgrade Tools, and is available on the Guidewire Resource Portal.
<i>New and Changed Guide</i>	Describes new features and changes from prior ClaimCenter 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 ClaimCenter. The intended readers are everyone who installs the application for development or for production.
<i>System Administration Guide</i>	Describes how to manage a ClaimCenter 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>PCF Reference Guide</i>	Describes ClaimCenter PCF widgets and attributes. The intended readers are configuration engineers.
<i>Data Dictionary</i>	Describes the ClaimCenter data model, including configuration extensions. The dictionary can be generated at any time to reflect the current ClaimCenter configuration. The intended readers are configuration engineers.
<i>Security Dictionary</i>	Describes all security permissions, roles, and the relationships among them. The dictionary can be generated at any time to reflect the current ClaimCenter configuration. The intended readers are configuration engineers.
<i>Globalization Guide</i>	Describes how to configure ClaimCenter 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 ClaimCenter.
<i>Rules Guide</i>	Describes business rule methodology and the rule sets in ClaimCenter Studio. The intended readers are business analysts who define business processes, as well as programmers who write business rules in Gosu.

Document	Purpose
<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 ClaimCenter 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 ClaimCenter 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>Java API Reference</i>	Javadoc-style reference of ClaimCenter Java plugin interfaces, entity fields, and other utility classes. The intended readers are system architects and integration programmers.
<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>Gosu API Reference</i>	Javadoc-style reference of ClaimCenter Gosu classes and properties. The reference can be generated at any time to reflect the current ClaimCenter configuration. The intended readers are configuration engineers, system architects, and integration programmers.
<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.
<b>bold</b>	Strong emphasis within standard text or table text.	You <b>must</b> define this property.
<b>narrow bold</b>	The name of a user interface element, such as a button name, a menu item name, or a tab name.	Next, click <b>Submit</b> .
<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.
<i>monospaced italic</i>	Parameter names or other variable placeholder text within URLs or other code snippets.	Use <code>getName(first, last)</code> . <a href="http://SERVERNAME/a.html">http://SERVERNAME/a.html</a> .

## Support

For assistance with this software release, contact Guidewire Customer Support:

- At the Guidewire Resource Portal – <http://guidewire.custhelp.com>
- By email – [support@guidewire.com](mailto:support@guidewire.com)
- By phone – +1-650-356-4955

---

## **part I**

# New and Changed in 9.0.0

This part includes information about ClaimCenter 9.0.0.



---

## chapter 1

# New and Changed in the Application in 9.0.0

This topic includes:

- “New in the Application in 9.0.0” on page 17
- “Changed in the Application in 9.0.0” on page 18

## New in the Application in 9.0.0

This topic includes:

- “Business Rules for Activities” on page 17
- “Straight-through Invoice Processing (STIP)” on page 18
- “Viewing Documents Associated with Vendor Contacts” on page 18

### Business Rules for Activities

Business rules for activity generation can be created and managed using the ClaimCenter Administration menu. This management tool is intended mainly for business analysts and administrators and enables dynamic configuration and deployment of business rules without system downtime.

#### See also

- “Business Rules for Activities” on page 519 in the *Application Guide*
- “Configuring Business Rules” on page 541 in the *Configuration Guide*

## Straight-through Invoice Processing (STIP)

Invoices meeting certain predefined criteria can be configured to be automatically approved and paid in ClaimCenter 9.0. This feature, also known as Straight-through Invoice Processing (STIP), can significantly accelerate the billing and payment workflow.

### See also

- “Configuring Straight-through Invoice Processing (STIP)” on page 696 in the *Configuration Guide*

## Viewing Documents Associated with Vendor Contacts

ContactManager enables you to attach documents to vendor contacts. For example, there might be a service level agreement that you require for vendors. Using ContactManager, you can add the agreement document to a vendor contact, and ContactManager then associates the document with that vendor.

If ClaimCenter is integrated with ContactManager, you can see read-only information about the documents associated with a vendor contact in the contact’s detail view. These documents remain associated with the vendor contact independently of the current claim. You cannot associate documents with vendor contacts independently of claims in ClaimCenter, but must log in to ContactManager for that purpose.

### See also

- “Vendor Document Management” on page 189 in the *Contact Management Guide*
- “Working with Contacts in ClaimCenter and ContactManager” on page 549 in the *Application Guide*
- “Document Management” on page 555 in the *Application Guide*

## Changed in the Application in 9.0.0

This topic includes:

- “Document Management Changes” on page 18
- “Subrogation Improvements” on page 19
- “Exposure Constraint Configuration” on page 19
- “Address field on Check” on page 19
- “Granular Recovery Management” on page 19
- “Authority Limits Configuration” on page 19
- “Financial Summary Screen Configuration” on page 19
- “Login Screen No Longer Saves User Information” on page 20

## Document Management Changes

The Document Assistant has been removed from the application because fewer browsers are supporting Java. As part of this change, there are new action buttons associated with documents, and there are more places that you can edit and upload documents. Additionally, it is now possible to upload more than one document at a time.

### See also

- “Document Management” on page 555 in the *Application Guide*

## Subrogation Improvements

ClaimCenter now allows you to manage subrogation at the exposure level. Subrogation is an assignable object. You can subrogate and assign individual exposures in a claim and also specify fault rating at this more granular level. Financial calculations for subrogation have been correspondingly adjusted as well.

**See also**

- “Subrogation” on page 267 in the *Application Guide*

## Exposure Constraint Configuration

In ClaimCenter, different exposures on a claim must have unique combinations of a claimant, coverage, and coverage subtype. In this release, this constraint is now configurable using the `ExposureDuplicateChecker` class. In addition, for liability exposures in claims involving third parties, the exposure uniqueness constraint is extended to include incidents.

**See also**

- “Exposures Screen” on page 35 in the *Application Guide*

## Address field on Check

The `MailToAddress` field in the `Check` entity, which contains the address of the person or company to whom the check is sent, was formerly a `String` value. In ClaimCenter 9.0, `Check` now contains `MailingAddress`, which is a foreign key reference to the `Address` entity. The `MailToAddress` field has been deprecated.

**See also**

- “Checks” on page 311 in the *Application Guide*

## Granular Recovery Management

ClaimCenter 9.0 enables the tracking of recoveries and recovery reserves by `Recovery Category`, which is now a required field. Consequently, recovery transactions erode recovery reserves appropriately from the specified recovery category.

**See also**

- “Recoveries and Recovery Reserves” on page 321 in the *Application Guide*

## Authority Limits Configuration

Authority limits can now be managed at a more granular level. Insurers now have the additional options of setting authority limits by policy type. It is also possible to add authority limit filter types such as lines of business, coverage subtype, cost category, and currency. Authority limits can also be extended to include additional limit types.

**See also**

- “Managing Authority Limit Profiles” on page 490 in the *Application Guide*

## Financial Summary Screen Configuration

The `Financial Summary` screen can now be configured to include additional `View` filter options as well as modify existing ones. Financial variants shown on this screen can be modified, and the hierarchy in which financial data is presented can also be altered. This configuration is done in Gosu.

**See also**

- “Configuring the Financial Summary Screen” on page 683 in the *Configuration Guide*

## Login Screen No Longer Saves User Information

The **Keep me logged in** check box on the **Login** screen has been removed, and ClaimCenter no longer stores a browser cookie that contains login information.

# New and Changed in Configuration in 9.0.0

This topic includes:

- “Changes to Configuration Files” on page 21
- “Changes to Configuration APIs” on page 22
- “Changes to the Data Model” on page 23
- “Changes to Guidewire Studio” on page 24
- “Changes to Configuration Parameters” on page 25
- “Changes Related to PCF Files” on page 26
- “Changes Related to Display Keys” on page 27

## Changes to Configuration Files

ClaimCenter 9.0.0 contains three configuration files related to security.

<code>security-config.xml</code>	Writable	Use to define custom security handlers.
<code>security-config-app.xml</code>	Read-only	Defines internal security permissions for the ClaimCenter application.
<code>security-config-p1.xml</code>	Read-only	Defines internal security permissions used by the underlying application code.

### Order of precedence

A permission declaration with the same entity and permKey values can appear in more than one of the files related to security. If a permission declaration appears in more than one file, the definition in `security-config.xml` takes precedence, followed by the definition in `security-config-app.xml`.

### New elements

Within file `security-config.xml`, the following XML elements are new:

- `<ObjectHandler>`
- `<OptionalObjectHandler>`

### See also

- “The Security Configuration File” on page 156 in the *System Administration Guide*

## Changes to Configuration APIs

This topic includes significant additions and changes to Gosu APIs, specifically, changes to core APIs in `gw.*`.

**Note:** Changes to query builder, integration, or web service APIs are described in other parts of the documentation, such as the *Integration Guide*.

This topic includes:

- “Changes to Exceptions in ClaimCenter” on page 22

## Changes to Exceptions in ClaimCenter

Parts of the exception hierarchy used in ClaimCenter have been simplified. As a result, some exceptions thrown in ClaimCenter code have been replaced with different types or exceptions in different packages. For example, code that used to throw `com.guidewire.pl.system.exception.IllegalFinancialsStatusException` now throws `java.lang.IllegalStateException` instead.

Only exceptions that were in `com.guidewire` packages are being changed, so it is not likely that this change affects your custom code. If you do have code that uses one of these exceptions, the exception code will longer work and at some point might cause a syntax error.

The following table shows the exceptions that are no longer used and the exceptions that replace them.

Previous Exception	Replacement Exception
<code>com.guidewire.pl.system.exception.IllegalFinancialsStatusException</code>	<code>java.lang.IllegalStateException</code>
<code>com.guidewire.pl.system.exception.IllegalTransactionException</code>	<code>com.guidewire.pl.web.controller.UserDisplayableException</code>
<code>com.guidewire.pl.system.exception.IllegalTransactionStatusException</code>	<code>java.lang.IllegalStateException</code>
<code>com.guidewire.pl.system.exception.EmptyTransactionSetException</code>	<code>com.guidewire.pl.web.controller.UserDisplayableException</code>
<code>com.guidewire.pl.system.exception.TransactionAmountChangeException</code>	<code>java.lang.IllegalStateException, prepended error message with "Transaction amount change"</code>
<code>com.guidewire.pl.system.exception.InvalidPaymentException</code>	<code>java.lang.IllegalStateException</code>
<code>com.guidewire.pl.system.exception.PaymentExceedsReservesException</code>	<code>gw.api.financials.PaymentExceedsReservesException</code>
<code>com.guidewire.cc.system.exception.CloseValidationException</code>	<code>gw.api.financials.CloseValidationException</code>

# Changes to the Data Model

This topic includes:

- “Changes to Data Model Entity Definitions” on page 23
- “Entity Types Can Declare Required Properties” on page 23
- “Typecode Identifier Customization” on page 24
- “Delegate Attribute requiresType” on page 24

## Changes to Data Model Entity Definitions

The following changes were made to the data model definitions:

Element	Change
<column>	<ul style="list-style-type: none"><li>• Added attribute required</li></ul>
<delegate>	<ul style="list-style-type: none"><li>• Added attribute requiresType</li><li>• Added attribute setterScriptability</li><li>• Removed attribute effdatedOnly</li><li>• Removed attribute requires</li><li>• Removed attribute subpackage</li></ul>
<forceindex>	<ul style="list-style-type: none"><li>• Removed attribute db2</li></ul>
<foreignkey>	<ul style="list-style-type: none"><li>• Added attribute archivingOwner</li><li>• Added attribute required</li><li>• Removed attribute owner</li></ul>
<implementsEntity>	<ul style="list-style-type: none"><li>• Removed attribute adapter</li><li>• Attribute impl is no longer required</li></ul>
<nonPersistentEntity>	<ul style="list-style-type: none"><li>• Removed attribute subpackage</li></ul>
<subtype>	<ul style="list-style-type: none"><li>• Removed attribute subpackage</li></ul>
<typekey>	<ul style="list-style-type: none"><li>• Added attribute required</li></ul>
<viewEntity>	<ul style="list-style-type: none"><li>• Added subelement &lt;implementsInterface&gt;</li><li>• Removed attribute subpackage</li></ul>

## Entity Types Can Declare Required Properties

You can now declare required properties for an entity type during object instantiation. To specify that a required entity property is required, set the attribute called `required` to the value `true`. The default is `false`. The `required` attribute is supported on the `<column>`, `<typekey>`, and `<foreignkey>` definitions of an entity type.

The `required` attribute does not overlap with the behavior of the existing attribute called `nullOk`. The `nullOk` attribute affects only the value of that property at commit time, which is the time the object is written to the database. In contrast, the `required` attribute affects only object instantiation. If the required property is an object type, the value must be non-null at the time of object instantiation or else Gosu throws an exception.

Code that uses the `new` operator to instantiate an entity type must include all required properties when constructing an instance of that entity as arguments to the constructor. If the constructor had other constructor arguments, the required properties appear after the other arguments.

During upgrade, this change may result in compilation errors in expressions that use the `new` operator to create entities with newly required properties that are omitted. To fix the compilation error, add the required fields after the other arguments.

For example, suppose an entity called `MyEntity` has one required parameter that is an `Address`. The code `new MyEntity()` is now a compilation error because the required property is missing from the argument list. To fix the error, get a reference to a non-null `Address` entity instance and pass it to the constructor, such as `new MyEntity(myAddress)`.

Some built-in entity types now have required fields. For new entity types, you can add your own custom required properties. However, you cannot add required fields to built-in entity types using data model extensions.

#### Required Bundle Arguments

As in previous releases, entity constructors contain an optional parameter that represents the bundle for the new entity instance. The value could be a bundle reference or another entity instance that is already in a writable bundle that you want to use.

If you provide the optional bundle parameter, the constructor argument list must include the bundle parameter before any required properties.

## Typecode Identifier Customization

ClaimCenter changed how typelist codes become Gosu programmatic identifier codes that appear as properties on the typelist type. For example, typecode abc becomes programmatic identifier code TC\_ABC.

The typecode definition element `<typecode>` has a new optional attribute called `identifierCode`. If you do not set `identifierCode`, ClaimCenter creates identifier codes using the same algorithm as in 8.0, but does not handle conflicts. If you set `identifierCode`, ClaimCenter creates the identifier by using the `identifierCode`. Use `identifierCode` to resolve identifier name conflicts in a way that is meaningful to you.

#### See also

- “Typecode Literals” on page 60 in the *Gosu Reference Guide*

## Delegate Attribute `requiresType`

There is a new feature in configuring entity delegates. To declare that the delegate must be implemented by an entity of a general type such as `retireable` or `versionable`, add the attribute `requiresType` on the `<delegate>` element.

Possible values for the `requiresType` attribute are the values of the `type` attribute on the `<entity>` element. See “Data Entities and the Application Database” on page 173 in the *Configuration Guide*. Some of the general types extend others. For example, `editable` extends `versionable`, and `versionable` extends `keyable`. An entity can implement the delegate if the implementing entity is the specified type or one that extends it. For example, if the `requiresType` attribute is `keyable`, then an implementing entity could have type `keyable`, `versionable`, or `editable`.

## Changes to Guidewire Studio

### Studio Editors Provide Editable XML

Many visual editors in Guidewire Studio—such as the Page Configuration editor or Typelist editor—also contain a tab that displays the XML code that defines the item shown in the editor. The label of this tab has changed from `XML` to `Text`. You can now also edit the XML code on the `XML` tab to make changes to the item. Valid changes to the XML code are automatically reflected in the visual editor. Invalid XML code may cause the visual editor to display the item incorrectly.

## Running Guidewire Studio

To run Guidewire Studio, run the command `gwb studio`. The script `studio` is no longer available.

## Changes to Configuration Parameters

See “Application Configuration Parameters” on page 35 in the *Configuration Guide* for a discussion of the parameters that you can use to configure Guidewire ClaimCenter.

This topic includes:

- “Removed Configuration Parameters” on page 25
- “Added Configuration Parameters” on page 25
- “Changed Configuration Parameters” on page 26

### Removed Configuration Parameters

The following configuration parameters have been removed in ClaimCenter 9:

- `AllowDocumentAssistant`
- `AllowSoapWebServiceReferenceNamespaceCollisions`
- `ClusterMulticastAddress`
- `ClusterMulticastPort`
- `ClusterMulticastTTL`
- `ClusterProtocolStack`
- `ClusterProtocolStackOption1`
- `ClusterProtocolStackOption2`
- `DocumentAssistantJNLP`
- `EnableClaimantCoverageUniquenessConstraint`
- `InputHelpTextOnFocus`
- `JGroupsClusterChannel`
- `JGroupsWatchdogHeartbeatIntervalSecs`
- `JGroupsWatchdogMissedHeartbeatsBeforeReset`
- `LoadSoapServicesOnStartup`
- `LoggerCategorySource`
- `LoggersShowLog4j`
- `LoggersShowPredefined`
- `SuppressIncidentExposureCoverageSubtypeEnforcement`
- `UseDocumentAssistantToDisplayDocuments`
- `UseSafeBundleForWebServiceOperations`
- `WizardNextShortcut`
- `WizardPrevNextButtonsVisible`
- `WizardPrevShortcut`
- `WSIEntityExpansion`

### Added Configuration Parameters

The following configuration parameters have been added in ClaimCenter 9:

Parameter	More information
<code>AddressDeletionDelay</code>	“ <code>AddressDeletionDelay</code> ” on page 56 in the <i>Configuration Guide</i>
<code>BizRulesDeploymentEnabled</code>	“ <code>BizRulesDeploymentEnabled</code> ” on page 44 in the <i>Configuration Guide</i>
<code>BizRulesDeploymentId</code>	“ <code>BizRulesDeploymentId</code> ” on page 44 in the <i>Configuration Guide</i>

Parameter	More information
BizRulesEnabled	"BizRulesEnabled" on page 43 in the <i>Configuration Guide</i>
BizRulesImportBootstrapRules	"BizRulesImportBootstrapRules" on page 44 in the <i>Configuration Guide</i>
BizRulesLeafSearchNumOfHops	"BizRulesLeafSearchNumOfHops" on page 44 in the <i>Configuration Guide</i>
ClaimOrExposureUnableToClosePattern	"ClaimOrExposureUnableToClosePattern" on page 71 in the <i>Configuration Guide</i>
DisableSequenceUtil	"DisableSequenceUtil" on page 52 in the <i>Configuration Guide</i>
FinalDocumentsNotEditable	"FinalDocumentsNotEditable" on page 55 in the <i>Configuration Guide</i>
MaxContactDocumentsSearchResults	"MaxContactDocumentSearchResults" on page 79 in the <i>Configuration Guide</i>
MaximumFileUploadCount	"MaximumFileUploadCount" on page 55 in the <i>Configuration Guide</i>
MaximumTotalUploadSize	"MaximumTotalUploadSize" on page 56 in the <i>Configuration Guide</i>
OnlineJavadocPrefix	"OnlineJavadocPrefix" on page 44 in the <i>Configuration Guide</i>
PrintCsvDefaultBaseFileExtension	"PrintCsvDefaultBaseFileExtension" on page 76 in the <i>Configuration Guide</i>
PrintCsvMimeType	"PrintCsvMimeType" on page 76 in the <i>Configuration Guide</i>
PrintDefaultBaseFileName	"PrintDefaultBaseFileName" on page 76 in the <i>Configuration Guide</i>
PrintPdfDefaultBaseFileExtension	"PrintPdfDefaultBaseFileExtension" on page 76 in the <i>Configuration Guide</i>
PrintPdfMimeType	"PrintCsvMimeType" on page 76 in the <i>Configuration Guide</i>

## Changed Configuration Parameters

The following configuration parameters have been changed in ClaimCenter 9:

Parameter	Change
ClusterMemberPurgeDaysOld	The minimum value is now 1. The parameter is now required.
DefaultApplicationCurrency	Can no longer be set per environment.
DefaultApplicationLocale	Can no longer be set per environment.
DefaultApplicationCurrency	Can no longer be set per environment.
DomainGraphKnownLinksWithIssues	Renamed to DomainGraphSafeLinks.
MigrateToLargeIDsAndDatetime2	Default changed from false to true.
PDFMergeHandlerLicenseKey	Can no longer be set per environment.
UISkin	Default changed from Titanium to theme-9.

## Changes Related to PCF Files

### Use Type-specific Input and Cell PCF Elements Instead of Generic Ones

The PCF elements Input and Cell are no longer allowed. Instead, use input and cell elements that are associated with particular value types. For example, Date Input, Text Input, Date Cell, Text Cell, and so on.

Accordingly, the new valueType property of an input or cell element is required. In most cases, valueType has a default value based on the type of element; for example, valueType for a Text Input defaults to `java.lang.String`. In cases where there is no default possible, such as for TypeKey Input or Range Input, you must set valueType explicitly.

## PCF Symbols are Case-sensitive

PCF elements, properties, and other symbols are now case-sensitive. Use the proper case when referring to them programmatically, such as in Gosu code or other configuration files.

## Other PCF Changes

The following are changes to the configuration of PCF files:

- The Search Panel element now requires a `searchCriteriaType` property.
- The Data Series and Dual Axis Data Series elements now require a `dataValueType` property.
- A Range Input that maps to multiselects must now set the `multiSelect` property to `true`.
- Iterator elements now require a `valueType` property.
- The Variable element now requires a `type` property, even when the `initialValue` property is set.

## Number and DateTime Types Have Been Replaced

To conform with Gosu type changes, the ClaimCenter make the following changes to types used in PCF files:

Removed type	Replace with type
Number	<code>java.lang.Double</code>
DateTime	<code>java.util.Date</code>

## PCF Changes Due to Upgrade

The configuration upgrade tools perform necessary modifications to PCF files.

## Changes Related to Display Keys

### Display Keys are Case-sensitive

Display keys are now case-sensitive. Use the proper case when referring to them programmatically, such as in Gosu code or other configuration files.

### Display Key Folder and File Structure is Flattened

In Guidewire Studio, the `configuration → config → Localizations` folder, which contained the `display.properties` files, no longer contains subfolders for each locale. Instead, the locale is now included in the file name, and all files are located at the top level of the `Localizations` folder. For example, the file `en_US/display.properties` is now named `display_en_US.properties`.

The underlying directory and file structure in the file system also reflects these changes.

### Access Display Keys with `DisplayKey.get`

The use of `displaykey.*` to refer to display keys in Gosu, PCF, and other configuration files is no longer valid.

To refer to display keys, use the method `DisplayKey.get`. For example:

Old reference	New reference
<code>displaykey.Web.Address.Default.PostalCode</code>	<code>DisplayKey.get("Web.Address.Default.PostalCode")</code>

You must include the following uses statement in any Gosu program that calls `DisplayKey.get`:

```
uses gw.api.locale.DisplayKey
```

# Changed in Globalization in 9.0.0

## Localizations Folder in Studio Has No Subfolders

In ClaimCenter 8.0, the **Localizations** folder in Guidewire Studio had subfolders for each location, such as `en_US`, `fr_FR`, `ja_JP`, and so on. Each of these location subfolders held property files, like `display.properties` and `localization.xml`, for a location,

In ClaimCenter 9.0, the **Localizations** folder has no subfolders. Each property file has a suffix to indicate the location. For string definition files for example, the `display_en_US.properties` file is directly under **Localizations** in 9.0, and is not in a subfolder.

### See also

- “Configuration Files Used for Globalization” on page 17 in the *Globalization Guide*



# New and Changed in Installation and Upgrade in 9.0.0

This topic includes:

- “Changed in Installation in 9.0.0” on page 31
- “New in Upgrade in 9.0.0” on page 31
- “Changed in Upgrade in 9.0.0” on page 33

## Changed in Installation in 9.0.0

This topic includes:

- “Apache Ant Installation Requirement Removed” on page 31

### Apache Ant Installation Requirement Removed

**Apache Ant is no longer required for general configuration and administration tasks.**

It is only necessary to install Ant if you are developing OSGi plugins using the IntelliJ IDEA with OSGi Editor.

#### See also

- “OSGi Plugin Deployment with IntelliJ IDEA with OSGi Editor” on page 648 in the *Integration Guide*

## New in Upgrade in 9.0.0

This topic includes:

- “Configuration Upgrade Tools” on page 32
- “Rolling Configuration Upgrade of ClaimCenter Cluster Members” on page 32

- “Custom Application Version Labels” on page 32
- “Upgrade-related System Tools Command Options” on page 33
- “Upgrade Flag Required for Upgrade” on page 33

## Configuration Upgrade Tools

The procedure for upgrading to ClaimCenter 9 has changed significantly. In ClaimCenter 9, there are new tools available for performing the configuration upgrade. Do not rely on your past knowledge of the upgrade process to begin the upgrade procedure without reviewing the new tools.

To upgrade your ClaimCenter Configuration, you must first download the configuration upgrade tools and documentation. The latest version of these tools and the *ClaimCenter Configuration Upgrade Guide* are available from the following location on the Guidewire Resource Portal:

[Guidewire InsuranceSuite Upgrade Tools page](#)

The configuration upgrade tools are:

**Pre-upgrade** – New command-line tool that prepares files for upgrade in the source configuration. The pre-upgrade tool modifies files in place in the starting version (ClaimCenter 7.0 or 8.0).

**Upgrade** – Updated version of the existing command-line configuration upgrade tool.

**Configuration Upgrade Merge Tracker** – New Guidewire Studio plugin used to track the status of files you are merging into your upgrade target version (ClaimCenter 9.0.0).

**Configuration Upgrade Smart Merge** – A new, syntactically aware, merge tool that merges Guidewire application configuration files. Smart Merge can be used from within Merge Tracker or from the command line.

For more information see the *ClaimCenter Configuration Upgrade Guide* available from the following location on the Guidewire Resource Portal:

[Guidewire InsuranceSuite Upgrade Tools page](#)

## Rolling Configuration Upgrade of ClaimCenter Cluster Members

It is now possible to update each application instance in a ClaimCenter cluster individually. Guidewire calls this type of upgrade a rolling upgrade or a configuration deployment. With a configuration deployment, it is possible to stop an individual ClaimCenter server and deploy configuration changes to it without stopping the other servers in the cluster. This type of upgrade is only possible with select application configuration changes.

### See also

- “Deploying Configuration Changes in a Clustered Environment” on page 137 in the *System Administration Guide*
- “Performing a Rolling Upgrade” on page 141 in the *System Administration Guide*

## Custom Application Version Labels

It is now possible to create custom application version labels that show in ClaimCenter, for example, in the **Server Tools Upgrade and Versions** screen. Use these version labels to identify a specific application version during an upgrade process.

### See also

- “Understanding Guidewire Software Versioning” on page 324 in the *System Administration Guide*

## Upgrade-related System Tools Command Options

There are new `system_tools` command options that relate to full (database) upgrade and rolling (configuration) upgrade. They are:

- `-cancelshutdown`
- `-completefailedfailover`
- `-components`
- `-evenifincluster`
- `-nodesfailed`
- `-nodes`
- `-requestcomponenttransfer`
- `-scheduleshutdown`
- `-shutdowndelay`
- `-startfullupgrade`
- `-terminatebatchprocesses`
- `-verifyconfig`

### See also

- “Administration Command Prompt Tools” on page 37
- “System Tools Command” on page 349 in the *System Administration Guide* for more information.
- “Deploying Configuration Changes in a Clustered Environment” on page 137 in the *System Administration Guide*

## Upgrade Flag Required for Upgrade

To prevent the unexpected upgrade of a ClaimCenter server, Guidewire requires that you set a ClaimCenter upgrade flag before starting the upgrade process:

- To start a full upgrade, you must first click **Start Full Upgrade** in the Server Tools **Upgrade and Versions** screen on any cluster member. At the completion of the upgrade, ClaimCenter removes the flag automatically.
- To start a rolling upgrade, you must first click **Start Rolling Upgrade** in the Server Tools **Upgrade and Versions** screen on any cluster member. After completing the upgrade of all servers in the ClaimCenter cluster, you must click **Rolling Upgrade Complete** on the **Upgrade and Versions** screen to remove the upgrade flag

### See also

- “Performing a Rolling Upgrade” on page 141 in the *System Administration Guide*
- “Unexpected Upgrades” on page 143 in the *System Administration Guide*
- “Upgrade and Versions” on page 322 in the *System Administration Guide*

## Changed in Upgrade in 9.0.0

This topic includes:

- “Changes to the Database Configuration File” on page 33

### Changes to the Database Configuration File

Guidewire has made the following changes to file `database-config.xml`:

- Automatic Database Upgrade Removed
- Index Statistics Element Removed
- Attribute `numbuckets` on Histogram Statistics Element Now Required

## Automatic Database Upgrade Removed

Guidewire has removed the `autoupgrade` attribute on the `<database>` element in `database-config.xml`. Instead, you must explicitly set an upgrade flag before starting either a full or rolling upgrade.

### See also

- “Upgrade Flag Required for Upgrade” on page 33

## Index Statistics Element Removed

Guidewire has removed the `<indexstatistics>` element from file `database-config.xml`. This change affects the following ClaimCenter functionality:

- The `system_tools -getdbstatistics`sstatements command no longer generates separate SQL statements for index statistics.
- The Server Tools **Database Catalog Statistics Information** download reports no longer contain statistics statements for indexes.

## Attribute `numbuckets` on Histogram Statistics Element Now Required

Guidewire now requires that you specify a value for attribute `numbuckets` on the `<histogramstatistics>` element in file `database-config.xml`. If you do not provide a value for this attribute, the ClaimCenter server generates an error and refuses to start.

# New and Changed in System Administration in 9.0.0

This topic includes:

- “New in System Administration in 9.0.0” on page 35
- “Changed in System Administration in 9.0.0” on page 37

## New in System Administration in 9.0.0

This topic includes:

- “Batch Process Load Balancing” on page 35
- “Messaging and Startable Server Load Balancing” on page 36
- “System Property Pass Through Option” on page 36
- “Graceful Server Shutdown” on page 36
- “New in Guidewire Server Tools” on page 36
- “New Address Deletion Work Queue” on page 36

### Batch Process Load Balancing

Guidewire now provides a default implementation of a lightweight prioritization strategy to use to load balance the running of batch processes within the ClaimCenter cluster. File `batch-process-config.xml` provides a way to control the execution of multiple batch processes.

#### See also

- See “Batch Process Load Balancing” on page 135 in the *System Administration Guide* for more information.

## Messaging and Startable Server Load Balancing

Guidewire now provides several default strategies for managing the load balancing of messaging destinations and startable services (plugins) within the ClaimCenter cluster. Use the following Gosu class to implement load balancing of these items.

```
DefaultBackgroundTaskLoadBalancingPlugin
```

### See also

- See “Messaging and Startable Service Load Balancing” on page 136 in the *System Administration Guide* for more information.

## System Property Pass Through Option

It is possible to specify Java system properties with the ClaimCenter command prompt tools using the following syntax.

```
-Dgw.passthrough.javaProperty=someValue
```

### See also

- “Using Pass-through Properties with the Command Prompt Tools” on page 42 in the *System Administration Guide*

## Graceful Server Shutdown

As there is no longer a single batch server, the order in which you shutdown the cluster servers becomes important.

### See also

- See “Cluster Member Shutdown” on page 121 in the *System Administration Guide* for information.

## New in Guidewire Server Tools

Guidewire has added the following server tool screens accessible to system administrators.

Screen	Description
Oracle Outlines	Guidewire has added an Oracle Outlines screen to the Info Pages of the ClaimCenter Server Tools. You use this screen to view information about stored outlines in the Oracle database. This screen is available only if the database is Oracle. See “Oracle Outlines” on page 312 in the <i>System Administration Guide</i> for more information.

## New Address Deletion Work Queue

During every bundle commit, ClaimCenter identifies addresses that are potentially orphaned as a result of that bundle commit. New work queue AddressDeleteWorkQueue is an internal work queue that ClaimCenter uses to delete orphaned addresses. New configuration parameter AddressDeletionDelay in file config.xml supports this new functionality.

### See also

- “Address Delete Batch Processing” on page 89 in the *System Administration Guide*
- “AddressDeletionDelay” on page 56 in the *Configuration Guide*

# Changed in System Administration in 9.0.0

This topic includes:

- “Usage Tools Command Removed” on page 37
- “Administration Command Prompt Tools” on page 37
- “Changes to Logging Infrastructure” on page 37
- “Modifications to the Server Registry Element in config.xml” on page 38
- “Removal of Support for JGroups Clustering” on page 38
- “Removal of Dedicated Batch Server Requirement” on page 38
- “Setting Server System Properties” on page 39
- “Changes in Guidewire Server Tools” on page 39

## Usage Tools Command Removed

Guidewire no longer supports the `usage_tools` command-line administrative tool and has removed the functionality from ClaimCenter.

## Administration Command Prompt Tools

The following list describes changes to the administration command prompt tools.

Command	Command options
<code>system_tools</code>	<p>Added the following command options:</p> <ul style="list-style-type: none"><li>• <code>-cancelshutdown</code></li><li>• <code>-completesafedfailover</code></li><li>• <code>-components</code></li><li>• <code>-evenifincluster</code></li><li>• <code>-nodefailed</code></li><li>• <code>-nodes</code></li><li>• <code>-requestcomponenttransfer</code></li><li>• <code>-scheduleshutdown</code></li><li>• <code>-startfullupgrade</code></li><li>• <code>-verifyconfig</code></li></ul> <p>Many of these new command options work specifically with upgrade, for example, <code>startfullupgrade</code>.</p> <p><b>See also</b></p> <ul style="list-style-type: none"><li>• “System Tools Command” on page 349 in the <i>System Administration Guide</i></li><li>• “Deploying Configuration Changes in a Clustered Environment” on page 137 in the <i>System Administration Guide</i></li></ul>
<code>table_import</code>	<p>Added the following command option:</p> <ul style="list-style-type: none"><li>• <code>-numthreadsintegritychecking</code></li></ul> <p><b>See also</b></p> <ul style="list-style-type: none"><li>• “Table Import Command” on page 355 in the <i>System Administration Guide</i></li></ul>
<code>usage_tools</code>	Guidewire has removed this command entirely.

## Changes to Logging Infrastructure

Guidewire has made the following changes to the ClaimCenter 9.0.0 logging infrastructure.

### Logging Parameters Removed

Guidewire has removed the following logging-related configuration parameters:

- `LoggerCategorySource`
- `LoggersShowLog4j`
- `LoggersShowPredefined`

This change affects the behavior of the Server tools **Set Log Level** screen. See “[Set Log Level](#)” on page 292 in the *System Administration Guide* for more information.

#### **Logger Class Deprecated**

Guidewire has deprecated class `gw.api.util.Logger`. Instead, use the following class to send information to application log files or the console window for debugging purposes:

```
gw.api.system.CCLoggerCategory
```

## **Modifications to the Server Registry Element in config.xml**

Guidewire has modified the server `<registry>` element in file `config.xml` so that:

- You define the valid and available server roles using the `roles` attribute on the `<registry>` element
- You define the environment, server ID, and roles assigned to a server using attributes on the `<server>` subelement of the `<registry>` element
- You define system properties for use with the servers in a ClaimCenter cluster using the `<systemproperties>` subelement of the `<registry>` element

#### **See also**

- “[Server Roles](#)” on page 119 in the *System Administration Guide*
- “[Using the Configuration Registry Element](#)” on page 38 in the *System Administration Guide*
- “[Using the Registry Server Element](#)” on page 40 in the *System Administration Guide*
- “[Using the Registry System Property Element](#)” on page 40 in the *System Administration Guide*

## **Removal of Support for JGroups Clustering**

Guidewire no longer supports JGroups clustering. Instead, ClaimCenter clusters now use point-to-point clustering. As a consequence, the following configuration parameters no longer exist in file `config.xml`:

- `ClusterMulticastAddress`
- `ClusterMulticastPort`
- `ClusterMulticastTTL`
- `ClusterProtocolStack`
- `ClusterProtocolStackOption1`
- `ClusterProtocolStackOption2`
- `JGroupsClusterChannel`
- `JGroupsWatchdogHeartbeatIntervalSecs`
- `JGroupsWatchdogMissedHeartbeatsBeforeReset`

#### **See also**

- “[Understanding ClaimCenter Server Clustering](#)” on page 117 in the *System Administration Guide*

## **Removal of Dedicated Batch Server Requirement**

In previous Guidewire ClaimCenter releases, each ClaimCenter cluster required a single server dedicated to batch processing. The other servers in the cluster handled various other tasks.

With the release of ClaimCenter 9.0.0, you can do the following:

- Assign one or more roles to each individual server in the ClaimCenter cluster
- Add custom server roles to those provided in the base configuration, which are `batch`, `messaging`, `scheduler`, `startable`, `workqueue`, `ui`
- Assign multiple servers the `batch` server role

**See also**

- “Modifications to the Server Registry Element in config.xml” on page 38
- “Understanding ClaimCenter Server Clustering” on page 117 in the *System Administration Guide*
- “Server Roles” on page 119 in the *System Administration Guide*

## Setting Server System Properties

Guidewire has modified the way that you set server system properties using the -D JVM option syntax. The exact command syntax depends on whether you are using the development QuickStart server (Jetty) or a production-level server such as Tomcat, for example:

Server	JVM option syntax
QuickStart (Jetty)	-Denv=... -Dserverid=...
Tomcat	-Dgw.cc.env=... -Dgw.cc.serverid=...

**See also**

- “Using the Registry Server Element” on page 40 in the *System Administration Guide*
- “Using the Registry System Property Element” on page 40 in the *System Administration Guide*.

## Changes in Guidewire Server Tools

Guidewire has made the following changes to the server tool screens accessible to system administrators.

Screen	Change
Batch Process Info	Guidewire has made changes to the way this screen presents information on batch processes and work queues. See “Batch Process Info” on page 284 in the <i>System Administration Guide</i> for details.
Cluster Info	Guidewire has renamed the Cluster Info screen to Cluster. From Cluster, you can navigate to a Cluster Member and a Cluster Component screen.  Previously, to view this screen, you had to enable clustering through the ClusteringEnabled configuration parameter in config.xml. This is no longer true. The clustering screens are visible to all with the proper logon credentials for Server Tools. See “Cluster Members and Components” on page 317 in the <i>System Administration Guide</i> for more information.
Loaded Gosu Classes	Guidewire no longer supports the functionality supplied by the Loaded Gosu Class screen, even though the screen still exists.
Set Log Level	Guidewire has removed the logging configuration parameters that controlled the behavior of this screen. Thus, it is no longer possible to configure this screen to show different hierarchies of logger categories. See “Set Log Level” on page 292 in the <i>System Administration Guide</i> for more information.
Startable Plugins	Guidewire has renamed the Startable Plugins screen to Startable Services. In addition, the screen summary information now includes the host on which the startable services is running. See “Startable Services” on page 317 in the <i>System Administration Guide</i> for more information.
Upgrade Info	Guidewire has renamed and modified the Upgrade Info screen, which now has the name Upgrade and Versions. The updated screen is no longer part of the Server Tools Info Pages screens. It is now its own separate category on the Server Tools screen. See “Upgrade and Versions” on page 322 in the <i>System Administration Guide</i> for more information.
Work Queue Info	Guidewire has slightly modified the name and functionality of the buttons that appear in the Actions column of the Work Queue table on the Work Queue Info screen. In addition, the Work Queue table now contains a Cluster-wide State column. See “Work Queue Info” on page 286 in the <i>System Administration Guide</i> for more information.



# New and Changed in Gosu in 9.0.0

This topic includes:

- “New in Gosu in 9.0.0” on page 41
- “Changed in Gosu in 9.0.0” on page 43

## New in Gosu in 9.0.0

This topic includes:

- “Structural Types” on page 41
- “Dynamic Types and Expando Objects” on page 42
- “Checked Arithmetic” on page 42

### Structural Types

Use structural typing to write code that works with objects with similar features but no common inheritance and interface declarations. Define structural types similar to defining interfaces, by specifying the common properties and method signatures. However, use the `structure` keyword, not the `interface` keyword.

Structural types are statically weaker than interfaces regarding the amount of enforced type information, but their flexibility supports situations where interfaces are ineffective or impossible. Structural types extend static typing to include a broader set of real world situations but still support concise code that catches common coding problems at compile time.

#### See also

- “Structural Types” on page 209 in the *Gosu Reference Guide*

## Dynamic Types and Expando Objects

Gosu supports dynamic language features that permit coding styles similar to non-statically typed languages. The `dynamic` type allows dynamic assignment as well as dynamic dispatch of property access and method invocation. Gosu also provides support for *expando objects*, which simplify dynamic property and method access for typical code contexts.

### See also

- “Dynamic Types and Expando Objects” on page 217 in the *Gosu Reference Guide*

## Checked Arithmetic

In version 8.0 and earlier versions, numeric values could exceed their defined bounds in arithmetic operations. For example, if you multiplied the maximum integer value by 2, the result by definition exceeds the range of integer values. Not only is the result value incorrect, the result could be positive when you expect it to be negative, or negative when you expect it to be positive. Because no Gosu exceptions occurred, it was difficult to protect against overflow errors, which could cause unexpected behavior or security issues.

In version 9.0, Gosu includes an optional feature called *checked arithmetic*. If checked arithmetic is enabled, Gosu behavior for the standard arithmetic operators changes for addition, subtraction, and multiplication. In nearly all cases, the result is the same and the behavior is the same. In the rare case that arithmetic overflow occurs, Gosu throws the exception `ArithmaticException`. To enable checked arithmetic, set Java system property `checkedArithmetic` to `true`.

Gosu checked arithmetic includes protection only for the operators `+`, `-`, and `*`. There is no protection for division, which only affects the expression `Integer.MIN_VALUE / -1`.

There are special cases in which arithmetic overflow behaviors are desirable for operators `+`, `-`, and `*`. For example, some common hash algorithms rely on arithmetic overflow. To handle typical use cases in overridden `hashCode` methods, Gosu always compiles `hashCode` methods with checked arithmetic disabled.

For other cases in which arithmetic overflow behaviors are desirable, you can use three new Gosu operators that ensure unchecked arithmetic independent of the Java system property `checkedArithmetic`. The new operators are the standard arithmetic operators prefaced with an exclamation point character: `!+`, `!-`, and `!*`.

For example, with Java system property `checkedArithmetic` to `true`:

```
var four = new Integer(4)
var y = Integer.MAX_VALUE * 2 + four // This line throws ArithmaticException
```

In contrast, the following example uses the unchecked arithmetic operator `!*`, which is only for special circumstances in which overflow is desirable:

```
var four = new Integer(4)
var x = Integer.MAX_VALUE !* 2 + four // This line does not throw ArithmaticException
print(x) // print "2"
```

Because the arithmetic in the second example is unchecked, the result that prints 2 successfully is possibly unexpected and invalid.

## JSON Support

Gosu has native support for JavaScript Object Notation (JSON) data format. JSON is an open-standard format that uses human-readable text to transmit data objects consisting of attribute-value pairs, hierarchical data structures, and arrays. Web sites often send or receive small amounts of JSON data as a lightweight alternative to the XML standard. Creation of JSON data and parsing can be implemented in any language or operating system, and is especially popular due to efficient web browser client-side parsing in JavaScript.

### See also

- “JSON” on page 325 in the *Gosu Reference Guide*

## Changed in Gosu in 9.0.0

This topic includes:

- “Entity and Typelist Case Sensitivity Changes” on page 43
- “Entity and Typelist Type System Changes” on page 44
- “Entity Instance Static Literal Syntax Removed” on page 45
- “Property Getter Methods Required for New Script Parameters” on page 46
- “Normalization of Gosu Code” on page 46
- “Metatype Properties and Methods No Longer Directly on the Type” on page 46
- “Number, DateTime, Bean, List Types Removed” on page 46
- “Find Expressions Removed” on page 47
- “Java Style Constructors Removed” on page 47
- “Implicit Coercions Removed” on page 47
- “XML API Changes” on page 49
- “GX Model Changes” on page 51
- “Gosu Property File Support Removed” on page 51
- “Any uses Lines Must Be At the Top of the File” on page 51
- “The typeis Operator Compile Time Changes” on page 51

### See also

- For changes in Gosu APIs that relate to the data model, see “Changes to the Data Model” on page 23.

### Entity and Typelist Case Sensitivity Changes

In version 8.0, the core Gosu language was case sensitive but types for entities and typelists remained case insensitive for many contexts.

In version 9.0, entity and typelist types are case sensitive in general for the following:

- Entity type names
- Entity method names
- Entity property names
- Typelist names
- Typelist value names

For most required changes, these differences cause compilation errors you can easily find and fix in Studio. For example, you might need to change `entity.address` to `entity.Address`. Fix all compilation errors.

Some required changes do not manifest as compilation errors but are still critical to change in version 9.0. If you pass any of the case sensitive names as `String` values to any API, it is best practice to ensure that all `String` values correctly capitalize the text. Otherwise, you see no compilation errors in Studio but your code generates run time errors that might be difficult or impossible to debug.

The following APIs changed to require case sensitive `String` values:

- Use of array syntax for entity property names: `object[PropertyName]` syntax. For example, `myObject["PrimaryContact"]`.
- Use of Gosu reflection APIs, such as methods of the class `gw.lang.reflect.ReflectUtil`.

Even though the property names are case-sensitive now, on an entity type you cannot create two entity properties with names that differ only by case. Similarly, in one typelist you cannot create two typecodes with names that differ only by their case.

## Some APIs Remain Case Insensitive

The following entity and typelist APIs remain case insensitive for `String` values:

- Database query API methods that take `String` values to identify fields
- Entity methods that get and set properties with a `String` value for the property name.
- Entity methods that check the contents of a bundle: `isFieldChanged`, `getOriginalValue`, `getAddedArrayElements`, `getChangedArrayElements`, `getRemovedArrayElements`, `isArrayElementChanged`, `isArrayElementAddedOrRemoved`. See “Determining What Data Changed in a Bundle” on page 342 in the *Gosu Reference Guide*.
- API methods that find a typekey by its code (as a `String`) is case-insensitive, for example:  
`typekey.TYPEREFTYPE.get("CODE")`.
- Entity path expressions (sometimes called bean path expressions) in view entity data model definitions remain case insensitive. See “View Entity Data Objects” on page 190 in the *Configuration Guide*. The following example shows an entity path in the `path` attribute:  
`<viewEntityName name="RelActAssignedUserName" path="RelatedActivity.AssignedUser"/>`

## Entity and Typelist Type System Changes

This topic includes:

- “Entity Supertype Behavior Changes” on page 44
- “In Type Information, Entity Types Are Never Final” on page 44
- “Typelist Types Are Not Java Enumerations” on page 45
- “Entity and Typelist Types Extend the IJavaType Interface” on page 45

### Entity Supertype Behavior Changes

ClaimCenter changed how Gosu handles supertypes in the entity type information property `Supertype`. This is the property accessed with an entity type such as `Address.Type.Supertype` or with an entity instance as `myObj.IntrinsicType.Supertype`.

In version 8.0:

- If an entity has a supertype, the `Supertype` property is the supertype entity as type `gw.entity.IEntityType`.
- If an entity has no supertype, the `Supertype` property is `null`.

In version 9.0:

- If an entity has a supertype, the `Supertype` property is the supertype entity as type `gw.lang.reflect.IType`.
- If an entity has no supertype, the `Supertype` property is `java.lang.Object`.

If you want the version 8.0 behavior, change your code to use the new property `Supertype_Entity` instead of `Supertype`. This new property is intended only for upgrading existing code. Do not use it for new code.

### In Type Information, Entity Types Are Never Final

ClaimCenter changed how Gosu handles entity finality from Gosu using the entity type information property `Final`. This is the property that you access with an entity type such as `Address.Type.Final` or with an entity instance as `myObj.IntrinsicType.Final`.

In version 8.0:

- If the data model definition declared an entity with the `final` attribute set to `true`, the `Final` property has the value `true`. Otherwise it has the value `false`.

In version 9.0:

- The `Final` property always has the value `false`.

If you want the version 8.0 behavior, change your code to use the new property `Final_Entity` instead of `Final`. This new property is intended only for upgrading existing code. Do not use it for new code.

## TypeList Types Are Not Java Enumerations

ClaimCenter typeList types are no longer directly equivalent to Java enumerations from Gosu APIs. This changes the behavior of the typeList type information property `Enum`. This is the property that you access with code such as `ContactType.Type.Enum`.

In version 8.0:

- The typeList type information property `Enum` returns `true`
- In method declaration values, you could define default values for typecode parameters, in other words parameters of typeList types. See “Named Arguments and Argument Defaults” on page 100 in the *Gosu Reference Guide*.

In version 9.0:

- The typeList type information property `Enum` returns `false`. Search your code for uses of the `Enum` property that depend on this behavior.
- In method declaration values, you cannot define default values for arguments of typecodes. If you want a similar behavior, define an additional method declaration that does not contain that typeList type argument. From the new method, call the original method and pass your original default typecode as an argument.

## Entity and TypeList Types Extend the `IJavaType` Interface

All entity and typeList types now extend the interface `gw.lang.reflect.java.IJavaType`. During upgrade, search your code for any code that checks if a type is an instance of `IJavaType`.

## Base Entities Are Removed

In version 8.0 and earlier releases, you could reference an entity type in Gosu using either its natural name (=as defined in the data dictionary) or with the `Base` suffix. In version 9.0, base entities are gone.

In regular code in Studio, any code that referenced a base entity is now a compiler error, and can be fixed by removing the `Base` suffix. For example, if any Gosu code references `AddressBase`, change that code to instead say `Address`.

You may have document templates or other Gosu templates that do not automatically show compiler errors in Studio. You must review your document template descriptors to find any Gosu code that uses entity names with the `Base` suffix.

## Entity Instance Static Literal Syntax Removed

In version 8.0, entity instance static literals using IDs were deprecated. In 9.0, this syntax is unsupported. Use query builder APIs instead. Use the ClaimCenter upgrade tools to update all existing code in Studio to the new style automatically.

Entity instance static literals with IDs look similar to object instantiation but do not use the `new` operator:

```
entityType(entityId)
```

Replace these expressions with query builder expressions such as:

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

### See also

- “Getting an Entity from a Public ID or a Key (Internal ID)” on page 340 in the *Gosu Reference Guide*

## Property Getter Methods Required for New Script Parameters

To make a new script parameter accessible from Gosu code, you must create a getter method for the new parameter in the script parameters enhancement file.

### See also

- “Working with Script Parameters” on page 120 in the *Configuration Guide*

## Normalization of Gosu Code

As part of automatic upgrade, ClaimCenter 9.0 reformats your Gosu code according to standard rules for spacing and whitespace. This process is called normalization. This process simplifies the other upgrade steps related to Gosu code, and makes your code easier to read.

## Metatype Properties and Methods No Longer Directly on the Type

Gosu exposes some type metadata as methods and properties associated with each Gosu type. These APIs are known as metatype methods and properties.

In version 8.0, methods such as `isAssignableFrom` or properties such as `Supertype` were moved to the `Type` property. The corresponding methods and properties directly on the type were deprecated.

In version 9.0, the deprecated APIs were removed and are now compilation errors. To fix the error, before the metatype property or method, add the word `Type` followed by a period character. Use the ClaimCenter upgrade tools to update all existing code in Studio to the new style automatically.

For example, this metatype method call worked but was deprecated in version 8.0:

```
String.isAssignableFrom(Number)
```

This is a compilation error in version 9.0. Instead, use the code:

```
String.Type.isAssignableFrom(Number)
```

## Number, DateTime, Bean, List Types Removed

The Gosu language no longer contains the types `Number`, `DateTime`, `Bean`, or `List`. In earlier releases, these were type aliases for other types. In version 9.0, you must use new names for these types. Use the ClaimCenter upgrade tools to update all existing code in Studio to the new style automatically.

The following table lists what types to use for any new code that you write

Removed type	Replace with type
<code>Number</code>	<code>java.lang.Double</code>
<code>DateTime</code>	<code>java.util.Date</code>
<code>Bean</code>	<code>Object</code>
<code>List</code> (when not qualified with a package)	For interfaces, replace with <code>java.util.List</code> . For classes, replace with <code>java.util.ArrayList</code>

### See also

- “Changes to List Instantiation” on page 46

## Changes to List Instantiation

Two changes happened to the `List` type in Gosu:

- The type `List` (with no package) is no longer a type alias for `java.util.List`. See “Number, DateTime, Bean, List Types Removed” on page 46.
- In new expressions, you cannot use the interface type `List` or `java.util.List` to instantiate a list as a shortcut to create an instance of `java.util.ArrayList`. If you get related compilation errors in new expressions, change your new expression to explicitly create an instance of `ArrayList` instead.

## Find Expressions Removed

The Gosu language no longer supports `find` expressions. Gosu retains equivalent functionality for retrieving information from the application database with the query builder APIs. Configuration upgrade tools for Version 9.0 convert legacy find expressions to equivalent query builder code.

For example, the `find` expression in following example Gosu code from earlier versions of ClaimCenter returns `Address` instances that are located in the city of Chicago.

```
// Query the database for addresses in Chicago.
var findQuery = find (a in Address where (a.City == "Chicago"))

// Print the addresses in the query result.
for (a in findQuery) {
    print (a.AddressLine1 + " / " + a.City + ", " + a.PostalCode)
}
```

After upgrade, the preceding `find` expression is converted to the following query builder code, which also returns addresses that are located in the city of Chicago.

```
uses gw.api.database.Query

// Query the database for addresses in Chicago.
var findQuery = Query.make(Address).compare("City", Equals, "Chicago").select()

// Print the addresses in the query result.
for (a in findQuery) {
    print (a.AddressLine1 + " / " + a.City + ", " + a.PostalCode)
}
```

### See also

- “Query Builder APIs” on page 127 in the *Gosu Reference Guide*

## Java Style Constructors Removed

In ClaimCenter 5.0, Java-style constructors were deprecated.

In version 9.0, Gosu does not support Java-style constructor declarations. For example, for a class called `ClassA`, instead of naming the constructor `ClassA`, use the Gosu native constructor name `construct`. Use the ClaimCenter upgrade tools to update all existing code in Studio to the new style automatically.

## Implicit Coercions Removed

All implicit coercions are eliminated, as well many explicit coercions using the `as` keyword. Different coercions have different types of replacements. The application automatically upgrades these coercions in your Gosu code as specified in the following table. Unless otherwise specified in the rightmost column, both implicit and explicit coercions are disallowed.

Previous coercion from this type	Previous coercion to this type	Upgrader replaces implicit coercions with this new Gosu code
<code>gw.pl.persistence.core.entity.KeyableB</code>	<code>gw.pl.persistence.core.Key</code>	<code>value?.ID</code>
<code>entity.ENTITYNAME[]</code>	<code>entity.TQuery</code>	<code>value.asQuery()</code>

Previous coercion from this type	Previous coercion to this type	Upgrader replaces implicit coercions with this new Gosu code
String	typekey.TYPERLIST	<p>For String literals in which the programming context defines it clearly as a typecode, the replacement is the code literal on the typelist type.</p> <p>For example, consider a variable of the type typekey.ADDRESSTYPE. To assigned the typekey code HOME:</p> <ul style="list-style-type: none"> <li>In 8.0 it was valid to assign the value "HOME". At compile time, Gosu converted it to the full value AddressType.TC_HOME if the programming context was the type ADDRESSTYPE.</li> <li>In 9.0, the new form is typekey.AddressType.TC_HOME or the more concise form TC_HOME.</li> </ul> <p>For non-constant values, the upgrade tool converts the implicit coercion to typekey.TYPERLIST.get(value).</p>
String	EncryptedString	new EncryptedString(value)
String	org.apache.commons.lang.enums.Enum	Direct reference to the constant on the enumeration class.
Numeric value	gw.api.financials.CurrencyAmount	gw.api.upgrade.Coercions.makeCurrencyAmountFrom(value)
T[]	java.util.List<T>	value.toList()
T[]	java.util.Set<T>	value.toSet()
java.util.Collection<T>	T[]	value.toTypedArray()
String	char[]	toCharArray()
String	String[]	new String[] { value }
Object	java.util.Date	gw.api.upgradeCoercions.makeTextFrom(value)
Object	String	<p>Implicit coercion is replaced by an explicit cast: value as String</p> <p>This is one of the few implicit coercions that can be replaced by an explicit coercion using the as keyword.</p>
Object	java.lang.StringBuffer	new StringBuffer(value as String)
Object	java.lang.StringBuilder	new StringBuilder(value as String)
Numeric values	Numeric values	<p>Replaced by method calls on the class gw.api.upgrade.Coercions. The method names have the form makeTYPEFrom. In some cases there are multiple methods for types that have Object and primitive versions. For example:</p> <ul style="list-style-type: none"> <li>For java.lang.Integer, the method is makeIntFrom.</li> <li>For the Gosu primitive int, the method name is makePIntFrom.</li> </ul> <p>This is one of the few implicit coercions that can be replaced by an explicit coercion using the as keyword. However, the behavior of as is different in rare edge cases. Thus, the upgrade tool uses a safer but more verbose upgrade transformation.</p>
<b>PolicyCenter business data coercions</b>		
String	T where T is option coverage term type	gw.api.upgrade.PCCoercions.makeOptionCovTerm<T>(value)

Previous coercion from this type	Previous coercion to this type	Upgrader replaces implicit coercions with this new Gosu code
	T where T is package coverage term type	gw.api.upgrade.PCCoercions. makePackageOptionCovTerm<T>(value)
	T where T is product model type	gw.api.upgrade.PCCoercions. makeProductModel<T>(value)

## XML API Changes

ClaimCenter changed the Gosu XML API. The changes you must make in your code to accommodate the changes depends on which XML classes you used and whether you used schemas (XSDs) to create Gosu types.

### XSD Type Code Generation

Support for XSD types are now implemented using code generation. After adding or changing any XSDs, you must regenerate the code generation with menu items in Studio, such as **Generate Xml Classes**.

#### See also

- “Regenerating XML Code Generation” on page 289 in the *Gosu Reference Guide*.

### Important Changes to Legacy XML API Based on XMLNode

There are important changes to the legacy XML API based on the `XMLNode` class:

#### Legacy XML API No Longer Loads XSDs

The legacy XML API (which is based on the `XMLNode` class) no longer supports loading an XSD and creating Gosu types from it. This change only affects XSDs that were registered to use the legacy XML API in ClaimCenter 8.0, which required an entry in the configuration file:

```
config/registry/compatibility-xsd.xml
```

If you added any lines to that configuration file, you must be aware of the change. Immediately convert your code to use the standard XML API based on `XmlElement`. See “Introduction to the XML Element in Gosu” on page 278 in the *Gosu Reference Guide*.

Because generated XSD types have a different structure with the current XML API, you may get compiler errors to fix the syntax of related code. For example, working with an arbitrary type called `MyXMLType`, the code `MyXMLType.Choice.ComplexType` must be change to `MyXMLType.ComplexType`.

For ClaimCenter, the `compatibility-xsd.xml` file referenced namespaces of a few XSD files in the default configuration. Any built-in code has been updated for this change. If you modified the built-in code to any code that uses types in the following packages, you may need to make changes to your code. Your new code must comply with the XSD type naming and `XmlElement` XML API:

- `xsd.acord`
- `xsd.xml-ns`
- `xsd.iso.req`
- `xsd.iso.resp`
- `xsd.iso.ak`
- `gw.xml.xsd.xsdtypes`

For example, the ClaimCenter integration with ISO used some of those packages. If you modified or wrote new code related to XSD types in the ClaimCenter integration with ISO, you may need to make changes to your code. In version 9.0, the ClaimCenter ISO integration now uses the XSD types as loaded by the standard XML API (`XmlElement`) not the legacy XML API (`XMLNode`).

### Legacy XML API Using Untyped XMLNode Instances Is Deprecated

The use of the `XMLNode` class as untyped XML nodes still works but is deprecated. Do not write new code that uses `XMLNode`. Begin to convert any `XMLNode` untyped node code to use the newer XML API based on the `XmLElement` class.

If you use the deprecated XML API, you might need to make manual changes due to minor package changes. In previous releases, the `XMLNode` class and the related interface `IXMLNode` both existed in two packages. In version 9.0, only the `gw.xml` package is valid for these types. Depending on which package you used, you might need to modify your code:

- If you used the class `gw.api.xml.XMLNode`, change all usages to `gw.xml.XMLNode`.
- If you used the interface `gw.api.xml.IXMLNode`, change all usages to `gw.xml.IXMLNode`.

### Changes to XmLElement If You Do Not Use XSDs

If you directly instantiate the `XmLElement` class rather than use schemas (XSDs) to create Gosu types, there are required changes.

Your code must change usages of special property names that now appear with a dollar sign (\$) prefix. To fix the compile errors, add the dollar sign prefix for the following `XmLElement` properties: `$AttributeNames`, `$Children`, `$DeclaredNamespaces`, `$MixedContent`, `$Namespace`, `$NamespaceContext`, `$QName`, `$SimpleValue`, `$Text`, `$TypeData`, and `$TypeInstance`.

For example, if your code with a compile error says `obj.Text`, change it to `obj.$Text`.

### Changes to XmLElement Using XSDs

Any reference to XML element attributes or properties called `Class` must change to `_Class`. If you get compile errors related to this change, add the underscore prefix.

### Customizing XSD Type Code Generation to Exclude Types

You can customize the way ClaimCenter converts XSDs to internal Java code, which is a process called *code generation*. By default, code generation happens for every XSD type. However, for better Studio performance, you can optionally omit code generation for some XSDs in some contexts or all contexts. The new `gwxmlmodule.xml` file controls this configuration.

#### See also

- “Customizing XSD Type Code Generation to Exclude Types” on page 295 in the *Gosu Reference Guide*

### Special Handling of XSD Enumerations with More Than 2000 Entries

If you use XSDs very large enumerations, there are changes that might affect your XML code that uses those XSDs. In ClaimCenter 9.0, if an XSD defines an enumeration (an `<xs:enumeration>` element) with more than 2000 entries, ClaimCenter converts the enumeration to a `String` value (an `<xs:string>` element).

This affects your Gosu code that uses the enumeration, as well as the overall type safety of related code. In addition to changing the type of any relevant properties, the Gosu compiler cannot perform compile-time verification of the individual enumeration values.

During upgrade, you might get compilation errors due to this change. Change the XML code that uses those properties by getting and setting `String` values instead of enumerations.

#### See also

- “Special Handling of Very Large XSD Enumerations” on page 296 in the *Gosu Reference Guide*

## XSD Types Always Have Lowercase Package Names

The generation of XSD types now always have lowercase package names in Gosu.

For example, suppose an XSD file is at the path `gsrc/example/MyTest.xsd` and contains an element called `Address`. In 8.0, the type was `example.MyTest.Address`. In 9.0, the type is `example.mytest.Address`. If you get Gosu compilation errors due to this change, fix the capitalization of the package names.

## XSD Substitution Group API Changes

The Gosu API changed slightly for the XSD feature called substitution groups. You may get compilation errors in some cases due to type incompatibilities. To fix the issue, call the `cast` method (with no arguments) on the type you want to assign. For example:

```
xml.Address = new schema.UKAddress().cast()
```

### See also

- “Substitution Group Hierarchies” on page 305 in the *Gosu Reference Guide*

## GX Model Changes

There are several changes in the Guidewire XML (GX) Modeler tool in Studio:

- After creating a new GX model or making changes to a GX model, you must generate code that lets you use this XML model in any Gosu code. In Studio, click **CodeGen** → **Generate Xml Classes**. Alternatively, you could regenerate all code generation types using the tool **CodeGen** → **Generate Everything**.
- For any GX model files, the product automatically upgrades all GX model source files to a new format in version 9.0. To identify the types within the model file itself, the new format has `xmltype` attributes on all property mappings. The product automatically determines the correct type. This change does not affect the API or usage of the tool in Studio. This change is mentioned to help understand file differences that you notice during product upgrade from an earlier release.
- If any models for a Gosu class contained a mapping for the `IType` property, that mapping is gone with no visible error. The `IType` property is no longer directly mappable. Alternatively, add and map a new Gosu enhancement property with similar information.

## Gosu Property File Support Removed

Native Gosu support for Java property files (`.property` files) is removed in 9.0.

The application automatically upgrades any code that used this API to equivalent but less concise code.

## Any uses Lines Must Be At the Top of the File

Any use of the `uses` keyword to include types must appear at the top of the file below the package declaration, but not within method declarations. In previous releases, `uses` lines were sometimes permitted in method declarations. The application automatically upgrades some configuration data that is implemented as Gosu classes, such as rule set data.

For other code contexts, fix any compilation errors that relate to this change.

## The `typeis` Operator Compile Time Changes

In version 9.0, the Gosu operator `typeis` is more restrictive in its use. The left side of the `typeis` operator must be an object of a compile-time type that is possible at run time to be the type specified after the operator. This is a new requirement that is enforced at compile time by the Gosu editor. For example, `typeis` works if the type on the right side is a subtype or supertype of the compile-time type of the left-hand side of the operator.

For example, the following code was valid in version 8.0 but is a compile error in version 9.0:

```
var b : Boolean = true
var x = (b typeis java.lang.Integer) // check if the type is a java.lang.Integer
```

The Gosu compiler knows that a Boolean object can never be a `java.lang.Integer`, so it is an error. This change increases the chance of detecting a certain class of accidental errors.

Generally speaking, any effects of this change result in compile errors but no run time errors. Fix any compilation errors that relate to this change.

If you use any dynamically-compiled code, test it or carefully review it for its use of the `typeis` operator. For example, the application dynamically compiles Gosu code embedded in templates used for Gosu document production.

## Annotations Changes

Annotations are a Gosu (and Java) syntax of adding metadata to a class declaration, or to part of a class, such as a method declaration. The `@` sign indicates usage of the annotation, for example `@WebService`. Some annotations take arguments.

In version 8.0, custom Gosu annotations were defined as classes that extend the interface `IAnnotation`. In version 9.0, the `IAnnotation` interface is deprecated, although any built-in annotations that use that annotation are not deprecated due to this change. Instead there is a new syntax for native annotations defined in Gosu using the `annotation` keyword. If you created any custom annotations with `IAnnotation`, begin to convert your annotations to the new syntax.

For the most part, using existing annotations from Gosu is unchanged. However, in version 8.0, native Java annotations required named arguments when used from Gosu, whereas Gosu `IAnnotation` annotations did not require named arguments. In version 9.0, argument names are now always optional for all annotations.

### See also

- “Annotations” on page 229 in the *Gosu Reference Guide*

# New and Changed in Integration in 9.0.0

This topic includes:

- “New in Integration in 9.0.0” on page 53
- “Changed in Integration in 9.0.0” on page 54

## New in Integration in 9.0.0

This topic includes:

- “SystemToolsAPI Web Service Has New Cluster Management Methods” on page 53

### SystemToolsAPI Web Service Has New Cluster Management Methods

The SystemToolsAPI web service has three new methods related to clusters:

- `getClusterState` – Returns a list of all nodes in the cluster, their roles, and what distributed components they run.
- `nodeFailed` – Cleans and releases resources (batch processes, plugins, and message destinations) reserved by a specified node.
- `completeFailover` – Completes component failover.

#### See also

- “System Tools Web Service” on page 96 in the *Integration Guide*

## Changed in Integration in 9.0.0

This topic includes:

- “ISO Integration Changes” on page 54
- “Changes to the ServiceRequestAPI Web Service” on page 54
- “MailToAddress Renamed in Financials Web Services DTOs” on page 55
- “RPCE Web Services Fully Removed” on page 55
- “WS-I Web Service Changes to regen-soap-api and @WsiGenInToolkit” on page 55
- “Document Management Changes” on page 55
- “Public ID Length Is 64 Characters” on page 58
- “Java Changes in 9.0.0” on page 58

### ISO Integration Changes

ClaimCenter ISO integration changes are as follows:

- For ISO support, ClaimCenter 9.0 requires use of the ISO ClaimSearch for DataPower Platform service, which is the newer version of the ISO ClaimSearch service. In ClaimCenter 8.0, DataPower support was optional and configured with the `ISO.DataPower` property in the `ISO.properties` file. If you have not yet coordinated with ISO to use the DataPower service, contact ISO Customer Service for their requirements and required SSL configuration. You will need to know your public key for the SSL certificate of the inbound callback ISO URL. For the settings for the `ISO.ConnectionURL` property in the `ISO.properties` file, see “ISO Properties File” on page 468 in the *Integration Guide*.
- Because DataPower support is now required, the `ISO.DataPower` property was removed and its value is now ignored.
- In version 9.0, it is required to use claim-based messaging with ISO for new claims. It is unsupported to use exposure-based messaging in the ISO integration. However, for claims with at least one exposure already sent to ISO (`Exposure.ISOSendDate` is non-null) using a previous version of ClaimCenter, that claim permanently uses exposure-based messaging. ClaimCenter continues to receive match reports for the exposures that were already sent to ISO. Also, for any updates to that claim, ClaimCenter send exposures to ISO as before. As in version 8.0, this behavior is configured with the `ISOClaimLevelMessaging` enhancement property implemented in the Gosu enhancement file `GWCclaimISOEnhancement.gsx`.
- In version 9.0, ClaimCenter removed the legacy XML API based on `Xm1Node`. Because the ClaimCenter integration with ISO ClaimSearch used the legacy XML API, related Gosu code now uses the newer XML APIs based on `Xm1Element`. If you made customizations to the ISO integration, you may need to make corresponding changes in your own code.
- The `ISOPreparePayload` class changed. This class no longer creates the request ID, but does still set the request ID (from a method argument) in the payload. If you changed this Gosu class, you must integrate these changes.

#### Also see

- “XML API Changes” on page 49
- “Gosu and XML” on page 277 in the *Gosu Reference Guide*.

### Changes to the ServiceRequestAPI Web Service

The ServiceRequestAPI web service changed as follows:

- The `addInvoice` method is now deprecated. Change your existing code to use the new method `addInvoiceToServiceRequest`
- There is a new method to add an invoice to a claim: `addInvoiceToClaim`

- The Amount field that was previous on the argument of type `StatementCreationInstructions` was replaced with an array of `ServiceRequestStatementLineItem` objects. Each `ServiceRequestStatementLineItem` object pairs an amount with fields that categorize and describe the amount.

## MailToAddress Renamed in Financials Web Services DTOs

The 9.0 versions of web services `ClaimFinancialsAPI` and `BulkInvoiceAPI` changed the property name for handling mailing addresses. The following data transfer objects (DTOs) had a `MailToAddress` property:

- `CheckDTO`, used with the `ClaimFinancialsAPI` web service
- `BulkInvoiceDTO`, used with the `BulkInvoiceAPI` web service

If you populated the `MailToAddress` property, change your code to use the new property name `MailingAddress`.

## RPCE Web Services Fully Removed

In version 8.0, ClaimCenter deprecated the older style of web services called RPCE.

In version 9.0, ClaimCenter removes RPCE web services from the product. Replace any RPCE web service code with WS-I web services.

As part of this change, the following data model attributes are no longer valid, because they were specific to RPCE web services: `exportable`, `exportasid`, and `SoapNullOk`. Because WS-I web services are not permitted to have arguments or return values that contain entity instances, these attributes have no meaning. Instead, design your web services to use data transfer objects (DTOs), either custom Gosu classes or XML/XSD types.

## WS-I Web Service Changes to regen-soap-api and @WsiGenInToolkit

In 8.0, most of the behavior of the `gwcc regen-soap-api` tool related to RPCE web services, which are no longer in the product. However, the tool also generated WSDL for locally published web services for web services with the `@WsiGenInToolkit` annotation. This tool added WSDL in the `ClaimCenter/soap-api` folder, not into the source code hierarchy.

In 9.0, the `gwcc regen-soap-api` tool is removed, including the WS-I behavior for local WSDL. In effect, there is no meaning to the `@WsiGenInToolkit` annotation in 9.0.

If you need the WSDL to test your web services, that feature is still available and generates the WSDL into the source code hierarchy in the `wsi.local` package. See “Testing Web Services with Local WSDL” on page 41 in the *Integration Guide*.

If you need the WSDL for external systems, you can still get the WSDL from a running server. See “Generating WSDL” on page 43 in the *Integration Guide*.

## Cluster Methods in MaintenanceToolsAPI Moved to SystemToolsAPI

Three methods relating to clustering moved from web service `MaintenanceToolsAPI` to web service `SystemToolsAPI`:

- `getClusterState`
- `nodeFailed`
- `completeFailover`

## Document Management Changes

This topic includes:

- “Document Assistant Removed” on page 56
- “Client-side Document Production Removed” on page 56

## Document Assistant Removed

In previous releases, the Guidewire Document Assistant provided interactive features to create, view, and edit documents. In version 9.0, there is no Document Assistant. Instead of enabling interactive viewing and editing of documents, ClaimCenter downloads documents to the local computer system through the download capabilities of the browser.

### See also

- For information on how to create, view, and edit documents without the Document Assistant, see “Document Production” on page 187 in the *Integration Guide*.

## Client-side Document Production Removed

In version 8.0, ClaimCenter provided client-side production of Microsoft Word and Microsoft Excel documents on Windows.

In version 9.0, client-side document production is removed entirely. However, there is now server-side production of Word and Excel documents. Guidewire strongly recommends using Word format instead of RTF for document production.

### See also

- For information on how to create template documents, see “Document Production” on page 187 in the *Integration Guide*.

## Document Upload Multiple File Upload and Editing

You can now upload multiple files at once. In the Actions menu, look for an action called **Upload Documents** or **Attach Documents**. You can drag multiple files from the local file system directly into the browser window.

To edit multiple files, navigate to the list of documents on an object. Click the checkbox at the top of the column to select all documents. Next, click the **Edit Details** button and follow the instructions on screen.

To control the maximum number of files to upload, modify the `config.xml` parameter `MaximumFileUploadCount`. To control the maximum size of a particular file, modify the `MaximumFileUploadSize` parameter. See “Configuration Parameters for Claim Document Management” on page 571 in the *Application Guide*.

## Document Upload Drag and Drop

Where there is a **Browse** button to find a local document to upload, you can drag files directly into the field next to the **Browse** button.

The user interface for multiple-file upload also supports drag and drop. See “Document Upload Multiple File Upload and Editing” on page 56.

## Document Template Descriptor Context Objects Can Refer to Earlier Context Object Symbols

In version 9.0, with document template descriptor definitions, Gosu expressions in context object definitions can refer to symbols from previously-defined context objects.

For example, suppose the XML file that defines a document template descriptor creates context objects in the following order: `Recipient1`, `Recipient2`, `Recipient3`. The definition for `Recipient2` can contain a Gosu expression that references the symbol `Recipient1`. Similarly, the definition for `Recipient3` can contain a Gosu expression that references the symbol `Recipient1` or `Recipient2`.

### See also

- “Document Template Descriptors” on page 199 in the *Integration Guide*

## Document Template Descriptor Supports Date Type

In version 9.0, with document template descriptor definitions, the context object type attribute supports an additional type called `date`. Specify that type for date values.

### See also

- “Document Template Descriptors” on page 199 in the *Integration Guide*

## Document Management Support in ContactManager for Vendor Documents

The ContactManager application now supports documents attached to vendor contacts only. Vendor document support is part of the integration with ClaimCenter and is not used in direct integrations with PolicyCenter or BillingCenter. For use with ClaimCenter, the ContactManager application supports document upload for vendor contact documents but does not support document production.

There is now an API to retrieve documents linked to a contact. In Studio, see the `ABContactAPI` class in the method `retrieveDocumentsForContact`.

### See also

- “DMS Servlet for Testing Vendor Documents in ContactManager” on page 57

## DMS Servlet for Testing Vendor Documents in ContactManager

ContactManager includes a new demonstration document management system implemented as a servlet. Because this code is intended for testing vendor documents in ContactManager, this code is implemented only in ContactManager not ClaimCenter.

The ContactManager servlet is called `DMSServlet`, which defines a RESTful API for use in testing a simulated DMS that returns URLs to vendor documents. In ContactManager, use the related document plugin implementation classes to simulate either a content store (with no metadata) or a complete DMS (content and metadata). You can configure the behavior using system properties.

**WARNING** Never use the `DMSServlet` code as a document content source in any production system.

If you want to simulate a DMS for testing ClaimCenter documents without a real DMS, you can register a separate document content source plugin implementation that save documents locally. See “Choices for Storing Document Content and Metadata” on page 158 in the *Integration Guide*. Those implementation classes are available in both ClaimCenter and ContactManager.

### The Document DTO Used by the DMS Servlet

To send and receive document metadata between the servlet and its clients, the DMS servlet and associated code uses a data transfer object (DTO) called the `<document>` XML element. This XML element is defined in the `DocumentDTO.xsd` file in the `gw.document` package. These XML objects are used only by the document management code associated with the `DMSServlet` REST APIs, which are intended for demonstration of a simulated DMS. Do not confuse the DTOs for the DMS servlet in the `DocumentDTO.xsd` file with similarly-named objects defined in Gosu classes and used by general purpose WS-I web services. For example, the ClaimCenter Gosu class `gw.webservice.cc.cc900.dto.DocumentDTO` is used by WS-I web services to represent a document.

### See also

- “Document Management Servlet for Testing Vendor Documents” on page 181 in the *Integration Guide*
- “Vendor Documents Overview” on page 190 in the *Contact Management Guide*
- “Choices for Storing Document Content and Metadata” on page 158 in the *Integration Guide*

## New Requirement for Implementations of IDocumentMetadataSource Plugin

If you write your own implementation of the `IDocumentMetadataSource` plugin interface, there is a required change during upgrade to version 9.0.

After retrieving a document, call the new `setRetrievedFromIDMS` method on the `Document` entity instance. The method sets `doc.PersistenceRequired` to `false` and marks the document as coming from the `IDocumentMetadataSource` plugin.

## New Event for Failure to Save Asynchronous Document

When a document fails to store in the database, ClaimCenter triggers a `FailedDocumentStore` event on the relevant `Document` entity instance. The event is important because it is the only notification that document storage failed. Use this event to create a notification for administrators or users.

### See also

- “Errors and Document Validation in Asynchronous Document Content Storage” on page 178 in the *Integration Guide*.

## Changes to APIs to Render Documents Directly to the Browser

You can display arbitrary `InputStream` content in a window, such as bytes of a PDF document that is encapsulated in a `DocumentContentsInfo` object.

In 8.0, you would use the following API:

```
gw.api.document.DocumentsUtil.renderDocumentContentsDirectly(fileName, docInfo)
```

In 9.0, that method is deprecated. Instead, call:

```
gw.api.document.DocumentsUtil.renderDocumentContentsWithDownloadDisposition(fileName, docInfo)
```

There are other new methods on the `DocumentsUtil` for rendering documents from file inputs. All the new methods start with the prefix `markFile`.

### See also

- “Retrieval and Rendering of PDFs, Input Streams, and File Widgets” on page 180 in the *Integration Guide*.

## PDF Production Plugin Parameter Removed

To configure PDF production, you must edit plugin parameters in the Plugins registry within the `IPDFMergeHandler.gwp` file. Although you still must edit that file to add the license number for the PDF production, the plugin parameter `BatchServerOnly` is no longer used. PDF document production now happens always on all servers.

### See also

- “Licensing for Server-side Document Production” on page 198 in the *Integration Guide*

## Public ID Length Is 64 Characters

In version 8.0, the maximum length of public IDs was 20 characters in ClaimCenter, BillingCenter, and ContactManager. PolicyCenter supported a maximum length of 64 characters.

In version 9.0, all Guidewire InsuranceSuite applications support 64 character public IDs.

## Java Changes in 9.0.0

ClaimCenter version 8.0 introduced the Guidewire Java API version 8.0. The Guidewire Java API version 7.0 was deprecated.

ClaimCenter version 9.0 introduces the Guidewire Java API version 9.0. The upgrade path for your code depends on your previous version:

- The Guidewire Java API version 7.0 remains deprecated but available in ClaimCenter 9.0. Guidewire strongly recommends that you upgrade your Java code to use the Guidewire Java API version 9.0 to avoid future upgrade issues. Additionally, Gosu code that accessed Java code that used the Java API version 7.0 was upgraded automatically for different default behavior. See “Changes to Gosu Code That Accesses Java API Version 7.0 Types” on page 60.
- The Guidewire Java API version 8.0 is removed and unavailable in ClaimCenter 9.0. You must upgrade your Java code from ClaimCenter 8.0 to use the Guidewire Java API version 9.0, or your code will not compile. See “Java API Changes for Entity and Typelist Data” on page 59.

Additional changes to the Guidewire Java API and Java configuration as described in the following topics.

### Java API Library Generation Changes

ClaimCenter changed the command line scripts that regenerate the Java API libraries and documentation.

To generate the libraries for the Java API version 9.0, at a command line prompt, change your working the application directory. Next, type the following command:

```
gwb genJavaApi
```

To generate the libraries for the deprecated Java API version 7.0, instead use the following command:

```
gwb genJavaApi -Ddeprecated=true
```

### Java API Reference Documentation Changes

The Java API Javadoc includes the specification of the plugin definitions for Java plugin interfaces, entity types, typelist types, and other types available from Java.

View the Java API reference at a new location:

```
ClaimCenter/javadoc/
```

The Javadoc contents are now static. The script that regenerates the Java API (gwb genJavaApi) does not regenerate the Javadoc. Therefore, your own data model changes are not reflected in the Javadoc documentation.

However, your changes to entity types, typecodes, and new properties are available from Java code in your Java IDE.

### Java API Changes for Entity and Typelist Data

The important differences between versions 8.0 and 9.0 of the Java API are primarily about accessing entity data and typelist data. The following table compares the Java API in both versions:

Java API feature	Usage in 8.0	Usage in 9.0
<b>Entity types</b>		
Java access entity types through what kind of Java type	Java interfaces	Java classes
Number of generated types for each entity type	1, 2, or 3, depending on the entity type and whether you extended the entity type	1
Same type name for Java entity types compared to Gosu	No, though it depended on which entity interface. Some interfaces has suffixes such as CoreExt or Ext.	Yes
Same package names for Java entity types compared to Gosu	No	Yes

Java API feature	Usage in 8.0	Usage in 9.0
Entity instantiation uses the standard new operator as in Gosu or typical Java code	No.  In version 8.0, you instantiated an entity using code such as <code>Address.TYPE.newInstance(b)</code>	Yes.  However, entity constructors from Java require a bundle argument, even if there is a current bundle. See “Create New Entity Instances from Java” on page 644 in the <i>Integration Guide</i> . In contrast, the bundle argument is optional in Gosu in contexts where there is a current bundle.
Access to Gosu enhancements from Java requires reflection APIs	Yes	Yes
<b>Typekey types</b>		
Typekeys	Yes	The typecode constants defined on the typelist classes now represent the actual instances (typekeys) of the typelist classes.
Getting an individual typecode from the typelist class.	To get a comparable instance of the typecode, you needed to use the typelist type get method.	Directly access instances from the typelist classes as static fields with the same names as before with the TC_ prefix. The typecode get method is no longer required, and is now deprecated.
You can use the == operator to compare two typekeys for equality.	No	Yes
Same package names compared to Gosu	No	Yes

### See also

- “Overview of Java and OSGi Support” on page 637 in the *Integration Guide*
- “Getting a Reference to an Existing Bundle in Java” on page 644 in the *Integration Guide*

### Changes to Gosu Code That Accesses Java API Version 7.0 Types

The Guidewire Java API version 7.0 remains deprecated but available. While your legacy code works in this release, Guidewire strongly recommends that you upgrade your Java code to use the version 9.0 API to avoid future upgrade issues.

Additionally, any Gosu code that accessed Java code that used the Java API version 7.0 was upgraded automatically to accommodate different default behavior. The result of the automatic upgrade is the same run time behavior as in 7.0 and 8.0.

In previous releases, ClaimCenter had two implicit behaviors when calling Java code that used the Java API version 7.0:

- For function arguments or return values, Gosu implicitly converted entity types to the external entities through a proxy.
- For function arguments or return values, Gosu implicitly cloned any collections.

Both behaviors no longer occur, so any code that used the Java API version 7.0 requires explicit conversion of entity types and explicit collection cloning. For existing code that you have not yet converted to Java API version 9.0, an automatic upgrade step converts any affected function calls. Your code is replaced by code that does those actions explicitly rather than implicitly. Although the code may be harder to read after conversion, the run time behavior is the same as in version 7.0 and 8.0. If you need to modify your code, Guidewire recommends upgrading your code to use the Java API version 9.0.

The configuration upgrade tools find all method calls to libraries and classes in the plugin directories that indicate the Guidewire Java API version 7.0. These method calls correspond to method calls on Java types that were defined in Java classes and libraries within plugin directories with paths that do not include /basic/.

The directories that trigger the code upgrade are:

- `ClaimCenter/plugins/PLUGINDIR/lib`
- `ClaimCenter/plugins/PLUGINDIR/classes`

The `PLUGINDIR` value represents a plugin directory as specified for that plugin implementation in the Studio Plugins editor. For use with general Gosu code not associated with a plugin, `PLUGINDIR` has the value `Gosu`. For code that applies to multiple plugin directories, `PLUGINDIR` has the value `shared`.

For any Gosu code that calls methods on those types, any Gosu in Studio is upgraded automatically. After the upgrade to version 9.0 is complete, there is no behavior difference between the plugin directory subdirectory that has `/basic/` in the path. See “Different Standard Location for Java Plugin Directories for non-OSGi Java Code” on page 61.

---

**IMPORTANT** After you upgrade any Java code to the Guidewire Java API 9.0, you can remove any of the added conversion code. However, remember that the behavior of collection is different in 9.0, because collections are not cloned as they were in the 7.0 version of the API. Before moving code to production, audit any related code that might be affected by this change in collection behaviors.

---

### Different Standard Location for Java Plugin Directories for non-OSGi Java Code

During upgrade to version 9.0, the location of Java files affects how Gosu code upgrades occur related to the deprecated Java API version 7.0. See “Changes to Gosu Code That Accesses Java API Version 7.0 Types” on page 60.

After upgrade to version 9.0 is complete, there is no behavior difference between plugin directory subdirectories with `/basic/` in the path compared to omitting that part of the path.

The old location works in this release but is now deprecated. Move your Java classes and libraries to the new standard location, which omits the `/basic/` in the path.

This change affects all classes and libraries in the directories:

- `ClaimCenter/plugins/PLUGINDIR/basic/classes`
- `ClaimCenter/plugins/PLUGINDIR/basic/lib`

The `PLUGINDIR` value represents a plugin directory as specified for that plugin implementation in the Studio Plugins editor. For use with general Gosu code not associated with a plugin, `PLUGINDIR` has the value `Gosu`. For code that applies to multiple plugin directories, `PLUGINDIR` has the value `shared`.

For example, move files in the following directory:

- `ClaimCenter/plugins/PLUGINDIR/basic/lib`

Move these files to the directory

- `ClaimCenter/plugins/PLUGINDIR/lib`

#### See also

- “Deploy Non-OSGi Java Classes and JARs” on page 647 in the *Integration Guide*



# New and Changed in 8.0 Maintenance Releases

This part includes information about ClaimCenter 8.0 maintenance releases.

**See also**

- “New and Changed in 8.0.0” on page 83



# New and Changed in ClaimCenter 8.0.5

**See also**

- For resolved issues and known issues in 8.0.5, see the release notes.

## Secure Transport (HTTPS) for Free-Text Search

You can configure ClaimCenter and the Guidewire Solr Extension to use the Hypertext Transfer Protocol Secure (HTTPS) communication protocol to exchange data. By default, ClaimCenter and the Guidewire Solr Extension use the unsecured HTTP protocol to exchange data. HTTPS provides a more secure exchange of data than HTTP by authenticating the ClaimCenter and Guidewire Solr Extension websites and associated web servers before connecting to exchange data. In addition, HTTPS provides bidirectional encryption of exchanged data.

**See also**

- “Configuring the Guidewire Solr Extension for Secure Transport (HTTPS)” on page 383 in the *Configuration Guide*

## WSS4J and xmlsec Library Changes

The built-in copies of WSS4J and xmlsec Java libraries are no longer available for customer use. If you want to use these Java libraries, include your own copies of the libraries. By using your own copy, you can also use older or newer versions of the library than the built-in versions.

**See also**

- “Overview of Java and OSGi Support” on page 637 in the *Integration Guide*



# New and Changed in ClaimCenter 8.0.4

This topic includes:

- “Changed in ClaimCenter 8.0.4” on page 67

**See also**

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

## Changed in ClaimCenter 8.0.4

This topic includes:

- “Template Tool Modified to Manage Email and Note Templates” on page 67
- “Bulk Purge Batch Processing” on page 67

### Template Tool Modified to Manage Email and Note Templates

Guidewire has modified the administrative tool command, `template_tools`, to work with email and note templates, as well as document templates. There are new commands to list and validate templates in all three categories.

**See also**

- “Template Tools Command” on page 358 in the *System Administration Guide*

### Bulk Purge Batch Processing

Bulk Purge batch processing has now been implemented as a work queue instead of a batch process. You can start a run of Bulk Purge processing from the **Work Queue Info** screen.

**See also**

- “List of Work Queues and Batch Processes” on page 87 in the *System Administration Guide*



# New and Changed in ClaimCenter 8.0.3

This topic describes the new and changed features in the ClaimCenter 8.0.3 maintenance release. For resolved issues and known issues in 8.0.3, see the release notes.

This topic includes:

- “New in ClaimCenter 8.0.3” on page 69
- “Changed in ClaimCenter 8.0.3” on page 69

## New in ClaimCenter 8.0.3

This topic includes:

## Changed in ClaimCenter 8.0.3

This topic includes:

- “Support for Oracle Partitioned Indexes” on page 69
- “Improving Claim Search Performance with Oracle Databases” on page 70
- “Change to Configuring ClaimCenter Exit Point to ContactManager” on page 70
- “Upgrade to Free-text Search” on page 70

### Support for Oracle Partitioned Indexes

ClaimCenter 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 38 in the *Installation Guide*

## Improving Claim Search Performance with Oracle Databases

If you are experiencing slow responses to claim searches that use claim contact criteria and your installation uses Oracle databases, there are two new indexes that can improve performance. The first index is for claim searches that include any contact role as a search criterion. The second index uses range partitioning on `ClaimantFlag` in the Oracle database to support claim searches that include a contact in the `Claimant` role. These two indexes are commented out in the new file `ClaimContact.etcx`. If you uncomment them, you must remove or comment out the index `C1mConLastFirstN`, which is also defined in that file.

**Note:** To uncomment these indexes and remove the existing index, you must open `ClaimCenter/modules/configuration/config/extensions/entity/ClaimContact.etcx` in a text editor. Because you are editing this file outside Studio, you must also open `extensions.properties` and increment the `version` attribute by 1. After saving the files, restart ClaimCenter to upgrade the database.

The two new indexes are:

```
<!-- Uncomment these indexes and remove the C1mConLastFirstN index if you are using Partitioning
<index desc="Covering index for search by person last name" expectedtobecovering="true"
       name="C1mConLastFirstNIn" trackUsage="true" unique="true">
    <indexcol keyposition="1" name="PersonLastNameDenorm"/>
    <indexcol keyposition="2" name="PersonFirstNameDenorm"/>
    <indexcol keyposition="3" name="Retired"/>
    <indexcol keyposition="4" name="ClaimID"/>
    <indexcol keyposition="5" name="ID"/>
</index>

<index desc="Covering index for search by person last name" expectedtobecovering="true"
       name="C1mConLastFirstNCl" trackUsage="true" unique="true">
    <indexcol keyposition="1" name="ClaimantFlag"/>
    <indexcol keyposition="2" name="PersonLastNameDenorm"/>
    <indexcol keyposition="3" name="PersonFirstNameDenorm"/>
    <indexcol keyposition="4" name="Retired"/>
    <indexcol keyposition="5" name="ClaimID"/>
    <indexcol keyposition="6" name="ID"/>
</index>
-->
```

**See also**

- “Configuring Index Partitioning for Oracle” on page 38 in the *Installation Guide*

## Change to Configuring ClaimCenter Exit Point to ContactManager

Previously, the web browser exit point from ClaimCenter to ContactManager used the URL defined in `suite-config.xml`, designed for ContactManager suite integration. For example, the `Edit in ContactManager` button, available in some ClaimCenter contact editing screens, uses this URL to open ContactManager in another web browser window. To provide separation between web browser and application integration URL definitions, this exit point is now defined in `config.xml` in the `ContactSystemURL` configuration parameter.

**See also**

- “Step 1: Integrate ContactManager with ClaimCenter” on page 45 in the *Contact Management Guide*

## Upgrade to Free-text Search

The Guidewire Solr Extension included with ClaimCenter 8.0.3 uses the Simple Logging Facade for Java (SLF4J) API. If you used free-text search with releases of ClaimCenter prior to 8.0.3, you must do the following:

- Modify the setup of the application server dedicated to the Guidewire Solr Extension
- Install the release of the Guidewire Solr Extension included with ClaimCenter 8.0.3.

No other configuration changes are required.

## Modifying Your Application Server Setup for Logging

You modify the setup of the application sever dedicated to the Guidewire Solr Extension by installing logging libraries included with ClaimCenter 8.0.3. Then, you configure the application server to make the logging libraries available. Logging JAR files are provided in the `logging_jars` folder at the root of the `cc-gwsolr.zip` file.

### See also

- For instructions to install the libraries and make them available, see the setup instructions for your application server type under “Free-text Search Setup” on page 86 in the *Installation Guide*.

## Installing Guidewire Solr Extension 8.0.3

1. Back up the existing index from your current installation of the Guidewire Solr Extension.
2. Uninstall your current version of the Guidewire Solr Extension from the application server.
3. Install the Guidewire Solr Extension included with ClaimCenter 8.0.3 from the distribution file `ClaimCenter/solr/cc-gwsolr.zip`.  
For complete instructions, see “Free-text Search Setup” on page 86 in the *Installation Guide*.
4. Restore the existing index from your backup location to your Guidewire Solr Extension hierarchy.
5. Start the Guidewire Solr Extension.
6. In a browser, open the administrative user interface for the Guidewire Solr Extension by entering the following URL:  
`http://hostName:8983/cc-gwsolr`
7. On the **Core Admin** page, click **Optimize** to upgrade the index files.



# New and Changed in ClaimCenter 8.0.2

This topic describes the new and changed features in the ClaimCenter 8.0.2 maintenance release. For resolved issues and known issues in 8.0.2, see the release notes.

This topic includes:

- “New in ClaimCenter 8.0.2” on page 73
- “Changed in ClaimCenter 8.0.2” on page 74

## New in ClaimCenter 8.0.2

This topic includes:

- “New Method to Import a Contact and Save It, Plus Two New Exceptions” on page 73
- “New Method to Save a Related ClaimCenter Contact in ContactManager” on page 74
- “New Plugin Interface InboundIntegrationMessageReply” on page 74
- “Data Change Menu Link” on page 74

### New Method to Import a Contact and Save It, Plus Two New Exceptions

From a Gosu class, you can now call a method that imports a contact by AddressBookUID from ContactManager. The new method in ClaimCenter, `importContactFromContactSystem`, is declared in the Java class `ContactSystemUtil`. The method has the following parameters:

- `java.lang.String abUIDOfContact` – The AddressBookUID of the remote contact to be imported.
- `boolean syncRelatedContacts` – If `true`, imports all related contacts and attaches them to the local contact. If `false`, does not import related contacts.

If successful, the method returns the locally created contact, which is linked to the remote contact.

The method can throw the following exceptions:

- `ContactSystemNotAvailableException` – A new exception that is thrown when the network connection to ContactManager from ClaimCenter fails.

- `ContactSystemContactNotFoundException` – A new exception that is thrown if the `AddressBookUID` specified in the call to `importContactFromContactSystem` is not found in `ContactManager`.
- `ContactSystemPluginException` – No contact system plugin is defined.

## New Method to Save a Related ClaimCenter Contact in ContactManager

When ClaimCenter is creating a contact in `ContactManager`, it now checks to see if any related contacts that were created along with the contact must also be created in `ContactManager`. To perform this check, ClaimCenter calls the new method `ContactSystemApprovalUtil.shouldCreateRelatedContactInContactSystem`. In the base configuration, this method returns `true` if the related contact is the primary contact for a vendor contact. You can configure the method to return `true` for other relationships, and you can add more restrictions as well.

## 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. For related upgrade information, see “[InboundIntegrationPlugin Plugin Interface Changed Name](#)” on page 75,

## 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 273 in the *System Administration Guide*

## Changed in ClaimCenter 8.0.2

This topic includes:

- “[Removed Configuration Parameters](#)” on page 74
- “[Integration Changes](#)” on page 74

## Removed Configuration Parameters

The following configuration parameters have been removed from ClaimCenter 8.0.2:

- `BucketSizeForHistogramsOnAllIndexedColumns`
- `CollectHistogramsOnAllIndexedColumns`

See “[The databasestatistics Database Configuration Element](#)” on page 191 in the *System Administration Guide*.

## Integration Changes

### New Data Change API

There is a new data change tool for rare cases of mission-critical data updates on running production systems. The usage is split into two separate tools intended for different users. One tool registers data change code. Another tool in the `Administration` user interface runs the code. Use the data change API under extraordinary conditions, with great caution, and only upon advice of Guidewire Customer Support. For important warnings and guidance, “[Data Change API Overview](#)” on page 273 in the *System Administration Guide*.

## 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.

## Web Service BulkInvoiceAPI Converted to WS-I

The RPCE web service `BulkInvoiceAPI` is now available as a WS-I web service. WS-I is a newer format of web service. See “Web Services Introduction” on page 27 in the *Integration Guide*. The new web service implementations are in the package `gw.webservice.cc.cc801.financials.bulkpay`.

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

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.

**IMPORTANT** 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`. Before editing these files, carefully review the introductory comments at the top of each DTO file.

There are important changes between the RPCE and WS-I versions:

- **removed method signature for creating bulk invoices** – The `createBulkInvoice` method signature with exchange rate and description no longer exists. The added data is on the new `BulkInvoiceDTO` object in properties `CustomExchangeRate` and `CustomExchangeRateDescription`.
- **Some method names changed in minor ways** – Some methods changed names slightly to improve clarity:
  - `addItems` is now called `addItemsToInvoice`
  - the three method signatures for the `getItems` method are now renamed to be more explicit: `getItemsForInvoice`, `getItemsForInvoiceAndClaim`, `getItemsForInvoiceAndClaimAndAmount`
  - `updateItems` is now called `updateItemsOnInvoice`
  - `deleteItems` is now called `deleteItemsFromInvoice`

- `createContactByLinkId` now uses the capitalization `createContactByLinkID`
- **for updated items, reversed meaning of null values** – the method `updateItems` had important changes. In the RPCE version, you had to be careful to always populate all values on the `BulkInvoice` SOAP entity. If any properties were `null`, the API set that value to `null`, potentially removing important data. In the WS-I version, by default all null property values are ignored. Only the `PublicID` property is required on the `BulkInvoiceDTO` object. As a consequence, by default you cannot set a value to `null` using this web service method. However, this is configurable. Go to the WS-I web service implementation class. At the end of the `updateItems` method, find the line:

```
itemDTO.writeTo(itemFromDB)
```

Change it to say:

```
itemDTO.writeTo(itemFromDB, false)
```

#### See also

- “Bulk Invoice Integration” on page 382 in the *Integration Guide*

# New and Changed in ClaimCenter 8.0.1

This topic describes the new and changed features in the ClaimCenter 8.0.1 maintenance release. For resolved issues and known issues in 8.0.1, see the release notes.

This topic includes:

- “New in ClaimCenter 8.0.1” on page 77
- “Changed in ClaimCenter 8.0.1” on page 78

## New in ClaimCenter 8.0.1

This topic includes:

- “Search for Claims by Claim Contact” on page 77
- “New Support for OSGi Plugins” on page 78
- “Gosu Using Clause Now Supports Finally Clause” on page 78

### Search for Claims by Claim Contact

In ClaimCenter 8.0.1, a new **Search by Contact** option provides faster, free-text search for claims than **Basic Search** or **Advanced Search**, which use relational database search. Free-text search is faster, because it searches through text-based representations of selected claim data instead of searching through multiple claim tables joined in complex relationships in the ClaimCenter application database.

The Guidewire free-text search feature that supports the **Search by Contact** option depends on a separate web application, the Guidewire Solr Extension. The Guidewire Solr Extension is a special distribution of Apache Solr, a full-text search engine, tailored by Guidewire to work with free-text search. Free-text search is disabled in the base configuration of ClaimCenter.

#### See also

- “Search by Contact” on page 68 in the *Application Guide*
- “Free-text Search Configuration” on page 373 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, ClaimCenter includes an application called IntelliJ IDEA with OSGi Editor separate from Guidewire Studio.

**See also**

- “Overview of ClaimCenter Plugins” on page 122 in the *Integration Guide*
- “Overview of Java and OSGi Support” on page 637 in the *Integration Guide*
- “OSGi Plugin Deployment with IntelliJ IDEA with OSGi Editor” on page 648 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 111 in the *Gosu Reference Guide*

## Changed in ClaimCenter 8.0.1

This topic includes:

- “Changes to Services” on page 78
- “Changes to Archiving” on page 78
- “Aggregate Limit Enhancements” on page 79
- “Conversion of Web Services from RPCE to WS-I” on page 79
- “Changes to Web Service Transaction IDs” on page 80
- “Change to Inbound Integration Configuration” on page 80
- “Gosu Suppress Warnings Annotation” on page 80
- “Gosu Compound Assignment Operators for Logical AND and Logical OR” on page 80

### Changes to Services

Service requests, like claims, exposures, and other assignable entities can now be assigned to a User in the Services screen.

**See also**

- “Assigning Service Requests” on page 394 in the *Application Guide*

### Changes to Archiving

In ClaimCenter 8.0.1, claims linked to a bulk invoice can be archived, even if they have activities or history events associated with them.

**See also**

- “Claims and Claim Entities that are not Archived” on page 286 in the *Application Guide*

## Aggregate Limit Enhancements

There is additional granularity available in viewing aggregate limit details. You can view claims contributing to aggregate limits by claim number and contributions broken down into specific details, such as the realized amount, net incurred contribution, and net paid contribution.

### See also

- “Aggregate Limits” on page 107 in the *Application Guide*

## 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*. The new web service implementations are in the package `gw.webservice.cc.cc801`.

**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.

**IMPORTANT** 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`. Before editing these files, carefully review the introductory comments at the top of each DTO file.

RPCE web service	Methods are now in the WS-I web service	Description
<code>IClaimAPI</code>	<code>gw.webservice.cc.cc801.claim.ClaimAPI</code>	Claim methods. See “Claim Web Service APIs and Data Transfer Objects” on page 100 in the <i>Integration Guide</i>
<code>IExposureAPI</code>	<code>gw.webservice.cc.cc801.claim.ClaimAPI</code>	Exposure-specific methods merged into the <code>ClaimAPI</code> web service along with claim-specific methods. See “Claim Web Service APIs and Data Transfer Objects” on page 100 in the <i>Integration Guide</i>
<code>IClaimFinancialsAPI</code>	<code>gw.webservice.cc.cc801.financials.ClaimFinancialsAPI</code>	Claim financials methods. See “Claim Financials Web Service (ClaimFinancialsAPI)” on page 359 in the <i>Integration Guide</i> .

Some important changes to the methods formerly in `ClaimAPI`:

- Methods that created documents are now encapsulated into a single method called `createDocument`. Properties on the data transfer object `DocumentDTO` define which secondary objects to associate the document with. See “Adding a Document from External Systems” on page 107 in the *Integration Guide*.

Some important changes to the methods formerly in `IClaimFinancialsAPI`:

- The WS-I web service APIs do not support recurring checks.
- The WS-I web service APIs that import or create financials no longer support grouped checks (multi-payee checks). However, the methods to void, stop, and reissue do support grouped checks. As before, the methods to apply foreign exchange adjustments do not support grouped checks.
- The methods `getExposureAmounts` and `getClaimAmounts` are not present in WS-I web services because most customers require their own versions that meet specific needs. Write your own new web services as appropriate for your specific external integrations.

In ClaimCenter 8.0.1, the `IBulkInvoiceAPI` web service continues to be provided in RPCE format only.

## 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 `@WsiCheckDuplicateExternalTransaction`. 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 Web Service Requests” on page 79 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 265 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 230 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 91 in the *Gosu Reference Guide*



---

## part III

# New and Changed in 8.0.0

This part includes information about ClaimCenter 8.0.0.

**See also**

- “New and Changed in 8.0 Maintenance Releases” on page 63



# New and Changed in the Application in 8.0.0

This topic includes:

- “New in the Application in 8.0.0” on page 85
- “Changed in the Application in 8.0.0” on page 86

## New in the Application in 8.0.0

This topic includes:

- “Services” on page 85
- “Changing the Screen Layout” on page 86

### Services

The Services feature is a major addition to ClaimCenter vendor support and enables adjusters to create service requests, gather quotes, and coordinate work with vendors. Service requests can be created during new claim creation or from the Actions menu. ClaimCenter manages service requests with vendors in conjunction with ContactManager and a communication portal for vendors such as the Guidewire Vendor Portal.

With the services feature, you can:

- Create service requests.
- Manage the progress of service requests with updates from the vendor.
- Add quotes, invoices, and other documents to the service request.
- Make payments using a built-in payment wizard when work is complete.
- Monitor the performance of different vendors using built-in metrics.

See also

- “Services” on page 387 in the *Application Guide*
- “Configuring Services” on page 519 in the *Configuration 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 63 in the *Application Guide*

## Changed in the Application in 8.0.0

This topic includes:

- “Special Handling – Customer Service Tiers” on page 86
- “Weighted Workload” on page 86
- “Multicurrency Reserving” on page 87
- “Aggregate Limits” on page 87
- “Changed Claim Exception Batch Processing” on page 87
- “Changes to the Address Book Tab” on page 88
- “Contact Changes No Longer Require Permissions” on page 88
- “Changes to Integration with ContactManager” on page 88

## Special Handling – Customer Service Tiers

Special handling refers to the additional procedures defined during claim processing for certain policies and claims, typically involving high-value customers. In ClaimCenter 8.0, the special handling feature has been extended to include *customer service tiers* associated with value-added services. Policies can be set up to belong to a customer service tier. When associated claims are created or modified, ClaimCenter checks rule triggers for special handling conditions and executes associated actions.

In the base configuration, the following customer service tiers are defined, based on the level of importance assigned to the policy holder:

- Platinum
- Gold
- Silver

The Silver customer service tier is not enabled in the base configuration.

### See also

- “Accounts and Service Tiers” on page 111 in the *Application Guide*
- “Configuring Special Instructions” on page 611 in the *Configuration Guide*

## Weighted Workload

Weighted workload assignment is a new enhancement to the assignment feature in ClaimCenter 8.0. It provides a more robust, intelligent, and configurable way to balance assignments than the previous round-robin method. You can now assign objects, such as claims or activities, based on a user’s current workload. The workload value is calculated based on the number and complexity of open objects owned by the user.

**See also**

- “Weighted Workload” on page 197 in the *Application Guide*

## Multicurrency Reserving

ClaimCenter 8.0 now provides complete support for reserving in multiple currencies.

With the enhanced multicurrency feature, you can:

- Designate a specific currency for a reserve line.
- Make payments, and have them erode reserves in the currency of the reserve line.
- Calculate financial totals accurately based on fluctuations in currency.
- View financials by currency type.
- Zero reserves appropriately based on reserving currency.

**See also**

- “Configuring Multicurrency” on page 631 in the *Configuration Guide*
- “Configuring Currencies” on page 117 in the *Globalization Guide*

## Aggregate Limits

The Aggregate Limits feature has been extended to include more granularity and flexibility in ClaimCenter 8.0.

Now, an adjuster can:

- Specify a coverage type, subtype, or a covered item that contributes to the limit.
- Archive and purge claims with associated aggregate limits.
- Display all the claims associated with a specific aggregate limit.
- Configure specific cost types and cost categories to include in an aggregate limit.
- Control the transactions that contribute to an aggregate limit.

The aggregate limit batch processes have also been enhanced.

**See also**

- “Aggregate Limits” on page 107 in the *Application Guide*
- “Understanding Aggregate Limits” on page 485 in the *Configuration Guide*
- “Batch Processes” on page 75 in the *System Administration Guide*

## Changed Claim Exception Batch Processing

Previously, the Idle Claim Exception batch processing ran claim exception for all open and closed claims that had not been changed for the configured `IdleClaimThresholdDays` number of days. In ClaimCenter 8.0, the Idle Claim Exception batch processing runs only against open, idle claims. A new type of batch processing, Idle Closed Claim Exception, can now be configured to run for closed, idle claims.

In the base configuration, Idle Claims Exception batch processing continues to be configured in the `scheduler-config.xml` file to run every Sunday at noon. The new Idle Closed Claim Exception batch processing is disabled (commented out). If you want to run Idle Closed Claim Exception batch processing, remove the comments and configure the schedule as needed.

## Changes to Document Assistant

The Document Assistant is now a Java applet installed by Guidewire applications to enable creating, viewing, editing, and uploading of files. The client machines require JRE7 or higher for the Document Assistant to run. The application user interface also includes some minor enhancements, as a consequence of this change.

## Changes to the Address Book Tab

The Address Book tab now supports only searching for contacts in an integrated contact management system, such as ContactManager. You cannot create, delete, or edit contacts directly in screens that open from the Address Book tab. As in previous versions of ClaimCenter, you can create new contacts, edit contacts, and delete them in screens associated with a claim, such as Parties Involved → Contacts.

There is a new **Open ContactManager** button available in the Address Book screens that you can use to open an instance of ContactManager. If you have a ContactManager user name with the appropriate contact and tag permissions, you can work with contacts directly in that application. Additionally, if you select a contact from the search results in the ClaimCenter Address Book search screen, you can click **Edit in ContactManager** to open ContactManager and edit the contact. You must log in separately to ContactManager when it opens.

## Contact Changes No Longer Require Permissions

In previous releases of ClaimCenter, a ClaimCenter user had to have the `abcreate`, `abedit`, and `abdelete` permissions to be able to create, edit, and delete contacts stored in ContactManager. Additionally, the user had to have permissions corresponding to the contact tag, like `anytagedit`. In this release, users can create and edit contacts stored in ContactManager without these permissions.

- Client contact changes are effective in ContactManager immediately, regardless of ClaimCenter user permissions.
- Vendor contact changes depend on user permissions, as follows:
  - If the user has the necessary contact and tag permissions, the changes that the user makes in ClaimCenter to a vendor contact are sent to ContactManager and take effect immediately.
  - If the user does not have contact and tag permissions, changes that the user makes in ClaimCenter to a vendor contact are sent to ContactManager as requests. ContactManager gives these requests Pending status. For the changes to become permanent, they have to be reviewed in ContactManager by a user who has permissions that enable reviewing and approving pending changes.

## Changes to Integration with ContactManager

See “New And Changed Features in ContactManager 8.0.0” on page 20 in the *Contact Management Guide* for more details on changes to integration.

These include:

- Gosu and XML Communication with ContactManager
- Plugin ClaimCenter implements from ContactManager web service for ContactManager to use
- Plugin to connect ClaimCenter to ContactManager
- Changes to *out of sync* determination for a ClaimCenter contact stored in ContactManager

# New and Changed in Configuration in 8.0.0

This topic includes:

- “Important Changes to the Configuration Module” on page 89
- “Changes to the Data Model” on page 90
- “Changes to the Generate Dictionary Command” on page 90
- “Changes to Guidewire Studio” on page 90
- “Changes to Configuration Parameters” on page 91
- “Changes to Web Services” on page 92
- “Changes Related to PCF Files” on page 92
- “Changes Related to Integrating with ContactManager” on page 94
- “Removal of Preload.txt Lines that Include Run Level” on page 95

## Important Changes to the Configuration Module

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

In prior versions of ClaimCenter, all your modified files were in the directory `ClaimCenter/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 `gwcc regen-dictionary` has changed with a parameter to generate XML versions of the *ClaimCenter Data Dictionary* and the *ClaimCenter 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 33 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 ClaimCenter 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 155 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 ClaimCenter 8.0.0:

- AllowActiveXAutoInstall
- CacheActive
- CreateNoteWithArchiveUpgradeIssues
- DisableHashJoinForProximitySearch
- DisableIndexFastFullScanForProximitySearch
- EnableExtRendering
- ExperimentalBrowserPerf
- FilteredArrayNoQueryThreshold
- FinancialsOverrides
- LegacyExternalEntityArraySupport
- UseDocumentNameAsFileName

### Added Configuration Parameters

The following configuration parameters have been added in ClaimCenter 8.0.0:

- ActivityPatternCacheMaxDuration
- AddIndexHintForLossDate
- AlwaysShowPhoneWidgetRegionCode
- ClusterMemberPurgeDaysOld
- ClusterMemberRecordUpdateIntervalSecs
- ClusterStatisticsMonitorIntervalMins
- ConsistencyCheckerThreads
- DefaultApplicationLanguage
- DefaultNANPACountryCode
- DefaultPhoneCountryCode
- DefaultXmlExportIEncryptionId
- DocumentAssistantJNLP
- EnableMultiCurrencyReserving
- JGroupsClusterChannel
- MaxAllowedDate
- MinAllowedDate
- PluginStartupTimeout
- ServiceRequestAPIMaxDaysKeepActiveWithoutInvoice
- ServiceRequestAPIMaxMessageResults
- ServiceRequestAPIMaxSearchResults
- VendorNotificationAPIRetryTime
- WeightedAssignmentEnabled
- WeightedAssignmentGlobalDefaultWeight
- WorkQueueThreadPoolMaxSize
- WorkQueueThreadPoolMinSize
- WorkQueueThreadsKeepAliveTime

## Changed Configuration Parameters

The following configuration parameters have been changed in ClaimCenter 8.0.0:

- `AllowActiveX` – Renamed to `AllowDocumentAssistant`, and default changed to `false`.
- `ClusterProtocolStack` – Default value changed.
- `InstantaneousContactAutoSync` – Default value changed to `true`. See “`InstantaneousContactAutoSync`” on page 69 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 “Existing WS-I Web Services Changes” on page 130.

## 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 ClaimCenter user interface users. The closest replacement for this context is a custom servlet. See “`Servlets`” on page 615 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 623 in the *Integration Guide*. Also see “`Template Overview`” on page 351 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, ClaimCenter 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 Integrating with ContactManager

For changes in ClaimCenter related to integrating with ContactManager and changes in ContactManager itself, see “New And Changed Features in ContactManager 8.0.0” on page 20 in the *Contact Management Guide*.

In particular, see the following table:

- “ClaimCenter File Name, Method, and Package Changes for ContactManager Integration” on page 25 in the *Contact Management Guide*

## Removal of Preload.txt Lines that Include Run Level

ClaimCenter 7.0.7 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 Globalization in 8.0.0

In ClaimCenter 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 ClaimCenter to a specific locale. In 8.0, the guide describes features for configuring Guidewire ClaimCenter to operate in a global environment.

This topic includes:

- “New in Globalization in 8.0.0” on page 97
- “Changes in Globalization in 8.0.0” on page 103

## New in Globalization in 8.0.0

The following globalization-related features are new in ClaimCenter 8.0:

- “Separation of Language and Regional Format” on page 98
- “Base Configuration Language Support” on page 98
- “Base Configuration Region Support” on page 98
- “Base Configuration Currency Support” on page 99
- “Base Configuration Geographical Support” on page 99
- “Phone Number Data and Phone Validator Gosu Class” on page 100
- “Automatic Address Completion and Fill-in Plugin Functionality” on page 100
- “Ability to Provide a Sort Order for the Typecodes in a Typelist” on page 101
- “Ability to Localize Script Parameter Descriptions” on page 101
- “Language Module Installation Utility” on page 101
- “Globalization-related Configuration Parameters” on page 102
- “Globalization-related Typelists” on page 102

- “Ability to Set First Day of Week by Region” on page 102
- “New in the ClaimCenter 8.0 Data Model” on page 102

## Separation of Language and Regional Format

New in ClaimCenter 8.0 is a separation between language and regional format. As part of the change in how ClaimCenter 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 ClaimCenter 8.0 with multiple languages or regional formats.

## Base Configuration Language Support

In the base configuration, Guidewire has a single default display language, English. If your installation has more than one language, you can select among the multiple languages by using the ClaimCenter **Options → Language** menu. ClaimCenter 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 → LocaleCode`

The term `LocaleCode` is a standard ISO locale code, such as `en_US` or `ja_JP`.

In a language module, each `localization.xml` file is in a `config/locale.LocaleCode` folder.

In the ClaimCenter 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, ClaimCenter 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 ClaimCenter formats dates, times, numbers, monetary amounts, and the names of people and companies.

**See also**

- “Working with Regional Formats” on page 87 in the *Globalization Guide*

## Base Configuration Currency Support

In ClaimCenter 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.

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

`configuration → config → currencies → currencyCode`

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

`configuration → config → currencies → aud`

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

**See also**

- “Configuring Currencies” on page 117 in the *Globalization Guide*

## Base Configuration Geographical Support

In ClaimCenter 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.xml` file into multiple `address-config` files, each stored under the `geodata` folder in its own folder named after the country code.

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

`configuration → config → geodata → AU`

**See also**

- “Configuring Address Information” on page 143 in the *Globalization Guide*

## New country.xml Files

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

ClaimCenter stores separate copies of `country.xml` under the `geodata` folder, each in its own `countryCode` folder.

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 143 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 provides multiple `zone-config.xml` files, each of which is used to define items like address autofill fields.

ClaimCenter stores the `zone-config` files under the `geodata` folder in individual `countryCode` folders.

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

`configuration → config → geodata → AU`

**See also**

- “Configuring Zone Information” on page 134 in the *Globalization Guide*

## Phone Number Data and Phone Validator Gosu Class

In 8.0, ClaimCenter 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 182 in the *Globalization Guide*

## Automatic Address Completion and Fill-in Plugin Functionality

Previous to 8.0, ClaimCenter used geographic zone definitions that you loaded into ClaimCenter to provide address autofill and autocompletion functionality. In 8.0, ClaimCenter 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 autfills 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 autfill 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 104.

#### See also

- “Address Autocompletion and Autofill” on page 152 in the *Globalization Guide*
- “Automatic Address Completion and Fill-in Plugin” on page 247 in the *Integration Guide*

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

New in ClaimCenter 8.0 is the ability to set the sort order (`priority` value) for the typecodes in a typelist by language. ClaimCenter 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. ClaimCenter 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 ClaimCenter 8.0, it is possible to localize the descriptions of script parameters. ClaimCenter 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 52 in the *Globalization Guide*

## Language Module Installation Utility

New in 8.0 is the ability to install a Guidewire-provided language module using the ClaimCenter `gwcc` 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*

## Globalization-related Configuration Parameters

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

- `DefaultApplicationLanguage`
- `DefaultPhoneCountryCode`
- `DefaultNANPACountryCode`
- `AlwaysShowPhoneWidgetRegionCode`

### See also

- “Globalization Parameters” on page 65 in the *Configuration Guide*

## Globalization-related Typelists

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

TypeList	Description
<code>LocaleType</code>	Use to define supported regions. The value of configuration parameter <code>DefaultApplicationLocale</code> must be a typecode from the <code>LocaleType</code> typelist.
<code>LanguageType</code>	Use to define supported languages. The value of configuration parameter <code>DefaultApplicationLanguage</code> must be a typecode from the <code>LanguageType</code> typelist.  Typelist <code>LanguageType</code> 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 <code>LanguageType</code> 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 ClaimCenter 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"> <li>• <code>en_AU</code></li> <li>• <code>en_CA</code></li> <li>• <code>en_US</code></li> <li>• <code>fr_CA</code></li> <li>• <code>ja_JP</code></li> </ul>
Monday	<ul style="list-style-type: none"> <li>• <code>de_DE</code></li> <li>• <code>en_GB</code></li> <li>• <code>fr_FR</code></li> </ul>

### See also

- For more information how ClaimCenter uses the ICU library, see “Locale Codes, `localization_localeCode.xml`, and the ICU Library” on page 88 in the *Globalization Guide*.

## New in the ClaimCenter 8.0 Data Model

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

- “New Phone Fields on Contact and Person Entities” on page 103
- “New Globalization-related Fields on Contact and Address Entities” on page 103
- “Attributes on `<localization>` Column Element” on page 103

## 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"><li>• FaxPhoneCountry</li><li>• HomePhoneCountry</li></ul>
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

## 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
unique	Boolean	If you set this attribute to true, ClaimCenter prohibits duplicate values in the set of value for each language.
nullok	Boolean	If you set this attribute to true, ClaimCenter flags missing entries that it finds during a database consistency check.  For example, suppose that ClaimCenter defines more than one language. In this case, ClaimCenter stores values for the default application language in the main database table of the entity. ClaimCenter stores the values for the additional languages in a separate localization table.  During consistency checks, ClaimCenter 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.

## Changes in Globalization in 8.0.0

This topic includes:

- “Changes to Configuration Parameter MulticurrencyDisplayMode” on page 104
- “Changes to Address Configuration 8.0” on page 104
- “Changes to Admin Data Localization” on page 104
- “Changes to 8.0 Typelists” on page 105
- “Changes to Files Used with Globalization” on page 106

## Changes to Configuration Parameter MulticurrencyDisplayMode

In ClaimCenter 8.0, the functional behavior of configuration parameter `MulticurrencyDisplayMode` changes compared to previous releases. Previously, ClaimCenter did not prevent changing `MulticurrencyDisplayMode` from `MULTIPLE` to `SINGLE`, but the documentation warned against doing so. In 8.0, Guidewire strictly enforces this constraint. 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

- “Setting the Currency Display Mode” on page 126 in the *Globalization Guide*

## Changes to Address Configuration 8.0

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

With ClaimCenter 8.0, Guidewire provides three PCF modes that cover the countries/regions that ClaimCenter 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 ClaimCenter. Guidewire strongly recommends that you convert any existing, pre-8.0 address entry PCF file to use the new `GlobalAddressInputSet` PCF.

ClaimCenter 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, ClaimCenter uses a new Gosu `NameFormatter` class to format names appropriately for internationalization.

### See also

- See “Configuring Address Information” on page 143 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 ClaimCenter, 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. ClaimCenter 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. ClaimCenter 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. ClaimCenter 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, ClaimCenter defines jurisdiction categories in this typelist, ClaimCenter 8.0 does not use any of the categories or types. Guidewire intends this typelist for use in future releases.
	<b>See also</b> <ul style="list-style-type: none"><li>• “Configuring Jurisdiction Information” on page 133 in the <i>Globalization Guide</i></li></ul>
State	Guidewire has modified the base configuration State typelist to include state definitions for the following countries: <ul style="list-style-type: none"><li>• AU</li><li>• CA</li><li>• DE</li><li>• JP</li><li>• US</li></ul>
	<b>See also</b> <ul style="list-style-type: none"><li>• “Configuring State Information” on page 133 in the <i>Globalization Guide</i></li></ul>
StateAbbreviation	Guidewire has added the StateAbbreviation typelist in 8.0. This typelist includes abbreviations for states in the following countries: <ul style="list-style-type: none"><li>• Australia</li><li>• Canada</li><li>• Germany</li><li>• United States</li></ul> 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.
	<b>See also</b> <ul style="list-style-type: none"><li>• “Configuring State Information” on page 133 in the <i>Globalization Guide</i></li></ul>

### See also

- “Configuring Geographic Data” on page 131 in the *Globalization Guide*
- “Working with Regional Formats” on page 87 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 ClaimCenter releases in which a file such as `localization.xml` contained data about more than one locale. In ClaimCenter 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
<ul style="list-style-type: none"> <li>• <code>address-config.xml</code></li> <li>• <code>country.xml</code></li> <li>• <code>zone-config.xml</code></li> </ul>	<p>Files <code>address-config.xml</code>, <code>country.xml</code>, and <code>zone-config.xml</code> 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:  <code>configuration → config → geodata → countryCode</code>  Substitute <code>countryCode</code> with a standard ISO country code.  In the base ClaimCenter 8.0 release, Guidewire provides support for the following countries:</p> <ul style="list-style-type: none"> <li>• AU</li> <li>• CA</li> <li>• DE</li> <li>• FR</li> <li>• GB</li> <li>• JP</li> <li>• US</li> </ul>
<code>currencies.xml</code>	<p>Contains currency code and similar information for the supported currencies. In single currency mode, currency is defined in <code>localization.xml</code> files under locale folders. See “Specifying Regional Formats for a Currency in Single Currency Mode” on page 127 in the <i>Globalization Guide</i>.</p> <p>In multiple currency mode, currency is defined in <code>currency.xml</code> files in currency folders, such as <code>usd</code>, located in the <code>currencies</code> folder. See “Specifying Regional Formats for Currencies in Multiple Currency Mode” on page 128 in the <i>Globalization Guide</i>.</p>
<code>config.xml</code>	<p>Globalization parameters new in ClaimCenter 8.0:</p> <ul style="list-style-type: none"> <li>• <code>DefaultApplicationLanguage</code></li> <li>• <code>DefaultPhoneCountryCode</code></li> <li>• <code>DefaultNANPCCountryCode</code></li> <li>• <code>AlwaysShowPhoneWidgetRegionCode</code></li> </ul> <p>Globalization parameters with modified functionality in ClaimCenter 8.0:</p> <ul style="list-style-type: none"> <li>• <code>DefaultApplicationLocale</code></li> <li>• <code>MulticurrencyDisplayMode</code></li> </ul> <p><b>See also</b></p> <ul style="list-style-type: none"> <li>• “Globalization Parameters” on page 65 in the <i>Configuration Guide</i></li> </ul>
<code>display.properties</code>	<p>There are now multiple <code>display.properties</code> files, one for each local folder. See “Localizing Display Keys” on page 47 in the <i>Globalization Guide</i>.</p>
<code>fieldvalidators.xml</code>	<p>You define national field validators in <code>fieldvalidators.xml</code> files located in country-specific folders in the <code>fieldvalidators</code> folder.</p> <p><b>See also</b></p> <ul style="list-style-type: none"> <li>• “Configuring National Field Validation” on page 181 in the <i>Globalization Guide</i>.</li> </ul>
<code>language.xml</code>	<p>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.</p> <p><b>See also</b></p> <ul style="list-style-type: none"> <li>• “Searching and Sorting in Configured Languages” on page 172 in the <i>Globalization Guide</i>.</li> </ul>

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 → LocaleCode</code></p> <p>For <code>LocaleCode</code> substitute a standard ISO locale code, such as <code>en_US</code>.</p> <p>In the ClaimCenter 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 ClaimCenter uses the ICU library defaults for that language.</p> <p><b>See also</b></p> <ul style="list-style-type: none"><li>“Configuring a <code>localization_localeCode.xml</code> File” on page 89 in the <i>Globalization Guide</i></li></ul>
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><b>See also</b></p> <ul style="list-style-type: none"><li>“Working with Phone Configuration Files” on page 166 in the <i>Globalization Guide</i></li><li>“Setting Phone Configuration Parameters” on page 166 in the <i>Globalization Guide</i></li></ul>

## 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 268 in the *Gosu Reference Guide*.



# New and Changed in System Administration in 8.0.0

This topic includes:

- “New in System Administration in 8.0.0” on page 109
- “Changed in System Administration in 8.0.0” on page 110

## New in System Administration in 8.0.0

This topic includes:

- “Phone Number Normalizer Work Queue” on page 109
- “User Workload Update Work Queue” on page 109

### Phone Number Normalizer Work Queue

ClaimCenter 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 ClaimCenter 7.0 single-field standard to the ClaimCenter 8.0 multi-field standard.

#### See also

- “Upgrading Phone Numbers” on page 121 in the *Upgrade Guide*

### User Workload Update Work Queue

ClaimCenter 8.0 includes a User Workload Update work queue with the weighted workload feature. This work queue updates workload data for ClaimCenter users. Any time a change is made that affects existing weighted workload values, run the User Workload Update work queue. In particular, any time classifications are modified, the changed classifications and any group users the classifications have been assigned to must be recalculated. Otherwise, the stored computed workload weights will be outdated.

**See also**

- See “Weighted Workload” on page 197 in the *Application Guide*.

## Changed in System Administration in 8.0.0

This topic includes:

- “Changes to Database Configuration” on page 110
- “Changes to Work Queue Configuration” on page 112
- “Changes to Scheduled Purge Batch Processing” on page 112
- “Changes to Database Statistics Updating” on page 112

### Changes to Database Configuration

As of ClaimCenter 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 ClaimCenter. 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 ClaimCenter. 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 ClaimCenter 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 ClaimCenter retries a failed work item for the work queue. If you do not specify a `retryLimit` value for a work queue, ClaimCenter uses the value of the `WorkItemRetryLimit` configuration parameter in `config.xml` as the default value.

ClaimCenter 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.

## ClaimCenter 7.0

Use `maintenance_tools` with one of the following options:

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

## ClaimCenter 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 `databasestatistics` 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 ClaimCenter 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 Gosu in 8.0.0

This topic includes:

- “New in Gosu in 8.0.0” on page 115
- “Changed in Gosu 8.0.0” on page 118

## New in Gosu in 8.0.0

### 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 ClaimCenter. To enable assertions, you must add the `-ea` flag on the JVM that hosts the application or Studio. For more details, see “Assert Statements” on page 112 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 is Now 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. For some types, this may not be useful.

However, if the constructor for the object triggers code that saves a copy of the new object, the return value from `new` may be unnecessary. This change may permit more concise code in some cases. See “New Is Optionally a Statement” on page 90 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 need to explicitly add uses lines to declare which annotation class to use. See “Annotating a Class, Method, Type, Class Variable, or Argument” on page 229 in the *Gosu Reference Guide*.

There are currently no compile time APIs for changing IDE behavior based on the annotations.

You can now access function parameter annotations using Java reflection APIs.

## 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 100 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, ClaimCenter 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

ClaimCenter protects entity instances from concurrent access through a version property that exists on all *versionable entities*. For details, see “Entity Instance Versioning and the Entity Touch API” on page 348 in the *Gosu Reference Guide*.

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 ClaimCenter 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.

This method existed in earlier versions of ClaimCenter, and was eventually deprecated. In ClaimCenter 8.0, the method is undeprecated. 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.

For more discussion about when to use this method, see “Entity Instance Versioning and the Entity Touch API” on page 348 in the *Gosu Reference Guide*.

## Changed in Gosu 8.0.0

### 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. See “Bundles and Database Transactions” on page 335 in the *Gosu Reference Guide*.

In previous releases, ClaimCenter represented a bundle in Gosu as the class `gw.transaction.Bundle`. The class still exists, but it is now in a different package. The new fully-qualified name is `gw.pl.persistence.core.Bundle`. Typical customers do not declare the package directly in their code, but instead 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 ClaimCenter 8.0 Java API. For details about this API, see “Java Code Changes” on page 131.

### 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> .

Map method name	Description
<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 true or false.
<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 true or false.

## 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(Claim).compare(Claim#PublicID, Relop.Equals, myPublicId).select().AtMostOneRow
```

For more information, see “Query Builder APIs” on page 127 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(Claim).compare(Claim#PublicID, Relop.Equals, myPublicId).select().AtMostOneRow
```

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

### See Also

- “Query Builder APIs” on page 127 in the *Gosu Reference Guide*
- “Getting an Entity from a Public ID or a Key (Internal ID)” on page 340 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 ClaimCenter, 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>
<code>com.guidewire.util.webservices.axis</code>	<code>gw.pl.util.webservices.axis</code>
<code>com.guidewire.util.webservices.login</code>	<code>gw.pl.util.webservices.login</code>
<code>com.guidewire.util.xml</code>	<code>gw.pl.util.xml</code>
<code>com.guidewire.util.xml.helpers</code>	<code>gw.pl.util.xml.helpers</code>
<code>com.guidewire.pl.plugin</code>	<code>gw.pl.plugin</code>
<code>com.guidewire.pl.plugin.addressbook</code>	<code>gw.pl.plugin.addressbook</code>
<code>com.guidewire.pl.plugin.approval</code>	<code>gw.pl.plugin.approval</code>
<code>com.guidewire.pl.plugin.credentials</code>	<code>gw.pl.plugin.credentials</code>
<code>com.guidewire.pl.plugin.custompicker</code>	<code>gw.pl.plugin.custompicker</code>
<code>com.guidewire.pl.plugin.dauth</code>	<code>gw.pl.plugin.dauth</code>
<code>com.guidewire.pl.plugin.document</code>	<code>gw.pl.plugin.document</code>
<code>com.guidewire.pl.plugin.management</code>	<code>gw.pl.plugin.management</code>
<code>com.guidewire.pl.plugin.messaging</code>	<code>gw.pl.plugin.messaging</code>
<code>com.guidewire.pl.plugin.search</code>	<code>gw.pl.plugin.search</code>
<code>com.guidewire.pl.plugin.security</code>	<code>gw.pl.plugin.security</code>
<code>com.guidewire.pl.plugin.util</code>	<code>gw.pl.plugin.util</code>
<code>com.guidewire.pl.plugin..validation</code>	<code>gw.pl.plugin.validation</code>
<code>com.guidewire.main</code>	<code>gw.pl.main</code>
<code>com.guidewire.external</code>	<code>gw.pl.external</code>

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>`. For more details on these topics, see “Concurrency” on page 375 in the *Gosu Reference Guide*.

### 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 “Concurrency” on page 375 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 `ClaimCenter/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 “Gosu Command Line Shell” on page 383 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 100 in the *Gosu Reference Guide*.



# New and Changed in Integration in 8.0.0

This topic includes:

- “New in Integration in 8.0.0” on page 123
- “Changed in Integration 8.0.0” on page 125

## New in Integration in 8.0.0

This topic includes:

- “New Phone Number Normalizer Plugin” on page 123
- “New MTOM Support for Results of Published WS-I Web Service” on page 124
- “Multi-threaded Inbound Integration” on page 124
- “New Messaging Destination Option for Messages Without Primary Object” on page 124
- “Regenerating WSDL and XSD in Web Service Collections (regen-from-wsc)” on page 125
- “New Plugin Interface IAggregateLimitTransactionPlugin” on page 125
- “New Plugin Interface PolicyLocationSearchPlugin” on page 125

### 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 247 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 ClaimCenter	Sending or receiving	Version 7.0 support	Version 8.0 support
ClaimCenter consuming a WS-I web service	Data with initial request	--	--
	Data in response	Yes	Yes
ClaimCenter 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 85 in the *Integration Guide*.
- For information about MTOM where Gosu is the SOAP server. See “Web Service Invocation Context” on page 39 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. ClaimCenter 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 ClaimCenter 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 265 in the *Integration Guide*

## New Messaging Destination Option for Messages Without Primary Object

In the ClaimCenter messaging system, many messages are associated with a primary object such as a claim. 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 323 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 by Using Web Service Collections in Studio” on page 68 in the *Integration Guide*

## New Plugin Interface IAggregateLimitTransactionPlugin

There is a new plugin interface `IAggregateLimitTransactionPlugin`. This plugin interface provides an optional capability to determine if a `Transaction` applies to an Aggregate Limit, beyond the usual values on the Aggregate Limit screen.

**See also**

- “Defining Aggregate Limits” on page 487 in the *Configuration Guide*

## New Plugin Interface PolicyLocationSearchPlugin

The new plugin interface `PolicyLocationSearchPlugin` performs a policy location search in a policy system such as PolicyCenter.

**See also**

- “Policy Location Search Plugin” on page 534 in the *Integration Guide*

**See also**

## Changed in Integration 8.0.0

This topic includes:

- “RPCE Web Services Deprecated” on page 126
- “WS-I Web Services That Replace Deprecated RPCE Web Services” on page 126
- “Existing WS-I Web Services Changes” on page 130
- “Java Code Changes” on page 131
- “Plugin Registry Changes in Studio” on page 137
- “Document Management Integration Changes” on page 137

- “ClaimCenter Contact Plugins Changes” on page 139
- “PolicyCenter Product Model Import into ClaimCenter Changes” on page 139

## RPCE Web Services Deprecated

ClaimCenter has two ways of publishing web services:

- **WS-I web services** – WS-I web services are published by using the SOAP protocol and are compatible with the WS-I standard.
- **RPC Encoded (RPCE) web services** – RPCE web services are an older style of publishing web services. Prior to version 7.0, ClaimCenter supported RPCE web service publishing only.

In version 7.0, Guidewire recommended you publish and consume web services as WS-I web services rather than RPCE. Version 7.0 continued to support RPCE web services, and RPCE web services in the default configuration were not deprecated.

In version 8.0, RPCE web services support and RPCE web services in the default configuration are deprecated. Version 8.0 continues to support RPCE web services for backward compatibility only. Guidewire strongly recommends you begin converting your RPCE code to equivalent WS-I code.

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 cc/800 or p1/800, that version was introduced in ClaimCenter 8.0.

### See also

- For summary information on changes to specific web services, see “WS-I Web Services That Replace Deprecated RPCE Web Services” on page 126.
- For information on writing code for WS-I web services, see “Web Services Introduction” on page 27 in the *Integration Guide*.

## WS-I Web Services That Replace Deprecated RPCE Web Services

RPCE web services in the base configuration are deprecated in version 8.0. Corresponding WS-I versions replace them. The replacements use different types for some method arguments and return values. For example, arguments and return values that took entity types in RPCE take corresponding types based on Gosu classes in WS-I. Replace your existing code that interacts with RPCE web services to interact with their WS-I replacements instead.

In some cases, WS-I web services replacements are named slightly differently than their corresponding RPCE web services. For example, WS-I names drop the initial letters “I” from their corresponding RPCE names. In addition, WS-I web services are in different packages than their corresponding RPCE web services, and WS-I package names include the version number of ClaimCenter.

This topic includes:

- “Import Tools Web Service” on page 127
- “Login Web Service” on page 127
- “Maintenance Tools Web Service” on page 127
- “Messaging Tools Web Service” on page 128
- “Profiling Web Service” on page 129
- “Systems Tools Web Service” on page 129
- “Template Web Service” on page 129
- “Typelist Tools Web Service” on page 130
- “Workflow Web Service” on page 130

## Import Tools Web Service

The WS-I web service `ImportToolsAPI` replaces the following deprecated RPCE web services:

- `IImportToolsAPI`
- `IUserAPI`
- `IGroupAPI`

### Replacement and New Methods for `IImportToolsAPI`

The `ImportToolsAPI` web service has replacements for the methods on `IImportToolsAPI`, as well as the following new methods:

- `importXml` – Imports only administrative data from XML data that conforms to an archiving import XSD or `tocc_import.xsd`
- `importArchiveXmlData` – Imports only administrative data from XML data in UTF-8 encoding that conforms to an XML format for archiving data
- `importArchiveXmlDataAsByteArray` – Imports only administrative data from XML data in a byte array that conforms to an XML format for archiving data

### Replacement Methods for `IUserAPI` and `IGroupAPI`

Use the `ImportToolsAPI` instead of `IUserAPI` and `IGroupAPI` to add a user or group with methods that import XML or CSV formatted text. For other methods in `IUserAPI` and `IGroupAPI`, there are no equivalent WS-I replacements. Guidewire recommends that you write your own WS-I web service to transfer only the necessary user or group data for each integration point.

#### See also

- For more information on the Import Tools web service, see “Importing Administrative Data” on page 89 in the *Integration Guide*.

## Login Web Service

The WS-I web service `LoginAPI` replaces the deprecated RPCE web service `ILoginAPI`. However, the WS-I version of `LoginAPI` differs notably from the RPCE version it replaces.

The deprecated RPCE web service `ILoginAPI` plays a critical role in RPCE for server authentication for subsequent method calls to the web service. If you use the generated Java libraries and its `APILocator` utility class, `APILocator` indirectly uses `ILoginAPI`.

The WS-I web service `LoginAPI` plays no analogous role in WS-I for server authentication. Authentication happens effectively with each WS-I method call. ClaimCenter provides the `LoginAPI` web service to let you test your web service client explicitly for specific authentication credentials. Call the `login` method with a username and a password as type `String`. If authentication fails, the `login` method throws an exception.

**Note:** You can also use the `LoginAPI` to intentionally leave a user session open for logging purposes.

#### See also

- “Login Authentication Confirmation” on page 59 in the *Integration Guide*

## Maintenance Tools Web Service

The WS-I web service `MaintenanceToolsAPI` replaces the deprecated RPCE web service `IMaintenanceToolsAPI`. In addition to the change from RPCE to WS-I, some methods were added, changed, or removed.

**Note:** The `MaintenanceToolsAPI` web service is available only if the system is at the `maintenance` run level or higher.

## New Methods

In Version 8.0, the following methods on the `MaintenanceToolsAPI` web service replace methods on the deprecated `IMaintenanceToolsAPI` web service:

### New Methods

The `MaintenanceToolsAPI` web service introduces the following methods for scheduling aggregate limits:

- `scheduleAggLimitRebuildOfClaims`
- `scheduleAggLimitRebuildOfPolicies`
- `scheduleAggLimitRebuildAllLimits`
- `scheduleAggLimitRebuildInvalidLimits`

The `MaintenanceToolsAPI` web services introduces the following methods related to archiving:

- `scheduleForArchive`
- `restore`

The `MaintenanceToolsAPI` web service introduces the `stopWorkQueueWorkers` method to stop query workers.

### Changed Methods

- `markPurgeReady` – Adds parameter. If the value is `true`, ClaimCenter purges each claim regardless of being part of an aggregate limit. If the value is `false`, ClaimCenter does not purge claims that are part of an aggregate limit but returns an error instead.
- `markPurgeReady` – Replaces the deprecated method `markForPurge`
- `getWQueueStatus` – Replaces the deprecated methods `getQueueWorkerStatus` and `getWorkQueueStatus` and acts on the entire cluster.

### Removed Methods

- `terminateBatchProcessByName` – Use `requestTerminationBatchProcessByName` instead.
- `terminateBatchProcessByID` – Use `requestTerminationOfBatchProcessByID` instead.

### See also

- “Maintenance Tools Web Service” on page 92 in the *Integration Guide*

## Messaging Tools Web Service

The WS-I web service `MessagingToolsAPI` replaces the deprecated RPCE web service `IMessagingToolsAPI`.

In Version 8.0, the WS-I web service `MessagingToolsAPI` has the following new methods not previously available with the deprecated `IMessagingToolsAPI` web service.

- `configureDestination` – New method to configure some performance and error handling aspects of a messaging destination on a running server.
- `getConfiguration` – New method to get performance and error handling parameters from a messaging destination on a running server.

Tools related to messaging, such as suspending a destination.

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`.

### See also

- “Messaging Tools Web Service” on page 346 in the *Integration Guide*

- “Messaging Tools Command” on page 346 in the *System Administration Guide*

## Profiling Web Service

The WS-I web service `ProfilerAPI` web service replaces the deprecated RPCE web service `IProfilerAPI`. In addition to the change from RPCE to WS-I, the following method signatures change:

- `setEnableProfilerForBatchProcess` – The method argument of type `BatchProcessType` now is `String`.
- `setEnableProfilerForWorkQueue` – The method argument of type `BatchProcessType` now is `String`.

### See also

- “Profiling Web Service” on page 95 in the *Integration Guide*

## Systems Tools Web Service

The WS-I web service `SystemToolsAPI` web service replaces the deprecated RPCE web service `ISystemToolsAPI`. In addition to the change from RPCE to WS-I, some methods were added or changed.

### New Methods

- `getDBCCState` – New method that returns the state of the consistency checks process
- `submitDBCCBatchJob` – New method that submits the consistency checks batch job on the underlying physical database
- `submitUpdateStats` – New method that runs the update stats process on the underlying physical database. Afterwards, use the `isUpdateStatsRunning` method to check on the process status.
- `cancelUpdateStats` – New method that cancels the update stats process if running. Afterwards, use the `getUpdateStatsState` method to check on the process status.
- `getUpdateStatsState` – New method that checks whether the update stats process is currently running.
- `submitAwrReportBatchJob` and `submitAwrReportBatchJobAllOpts` – New methods to submit Oracle AWR Performance Report batch jobs
- `getPerfReportsInfo` – New method to retrieve information about the latest Oracle AWR downloads

### Changed Methods

- `submitDmvReportBatchJobWithParams` – The now takes the single `includeDatabaseStatistics` argument instead of three arguments
- `getRecentAWRSnapshotInfo` – The method now returns `DatabaseSnapshotInfo[]` instead of `String`
- `updateLoggingLevel` – Logging level now is passed as `LoggingLevel` instead of `String`

### See also

- “System Tools Web Service” on page 96 in the *Integration Guide*

## Template Web Service

The WS-I web service `TemplateToolsAPI` web service replaces the deprecated RPCE web service `ITemplateToolsAPI`. In addition to the change from RPCE to WS-I, the methods were added:

- `validateTemplateInLocale` – New method to validate one template using a specific locale code
- `validateAllTemplatesInLocale` – New method to validate all templates using a specific locale code

### See also

- “Template Web Service” on page 213 in the *Integration Guide*

## TypeList Tools Web Service

The WS-I web service `TypeListToolsAPI` web service replaces the deprecated RPCE web service `ITypeListToolsAPI`.

### See also

- “Using Web Services to Translate Typecodes” on page 88 in the *Integration Guide*

## Workflow Web Service

The WS-I web service `WorkflowAPI` web service replaces the deprecated RPCE web service `IWorkflowAPI`. In addition to the change from RPCE to WS-I, the following method signatures change:

- `invokeTrigger` – The method argument of type `WorkflowTriggerKey` now is `String`.
- `isTriggerAvailable` – The method argument of type `WorkflowTriggerKey` now is `String`.

### See also

- “Workflow Web Service” on page 98 in the *Integration Guide*.

## Existing WS-I Web Services Changes

- “Contact Web Service Changes” on page 130

### Contact Web Service Changes

The `ContactAPI` web service lets an external system manipulate contact data in ClaimCenter. `ContactManager` calls this web service.

#### Transaction IDs No Longer Used

In version 7.0, some methods took a transaction ID, which represented a unique identifier for an update request. If a request is duplicated later, ClaimCenter identifies it as duplicate by its transaction ID and does not repeat the request.

In 8.0, methods no longer take transaction IDs. Instead, they have the annotation `@WsiCheckDuplicateExternalTransaction`, which automatically enforces unique transaction IDs based on SOAP headers. This change affects the following methods:

- `updateContact`
- `removeContact`
- `mergeContacts`

If the web service client is another Guidewire application, you can set the transaction ID with the method `ContactAPIUtil.setTransactionId`.

#### New Methods for Address Book Client API

The `ABCClientAPI` published by `ContactManager` 8.0 has the following new methods to help manage changes to contact information:

- `pendingUpdateApproved`
- `pendingCreateApproved`
- `pendingUpdateRejected`
- `pendingCreateRejected`

These methods are intended as callbacks from the ContactAPI web service that the InsuranceSuite applications PolicyCenter, BillingCenter, and ClaimCenter publish for ContactManager to call. However, each application differs in how its implementation of the ContactAPI web service uses the methods call back to the ABCClientAPI that ContactManager publishes. ClaimCenter For pendingCreateApproved, the application simply logs the request. For pendingUpdateApproved, ClaimCenter updates the contact graph entities that the update operation created. The AddressBookUID values need to be updated with the values that ContactManager assigns. For pendingUpdateRejected and pendingCreateRejected, ClaimCenter creates new activities.

#### See also

- “Checking for Duplicate External Transaction IDs” on page 51 in the *Integration Guide*
- “Contact Web Service APIs” on page 502 in the *Integration Guide*

## Java Code Changes

ClaimCenter 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 ClaimCenter 8.0.
- To regenerate the libraries for the legacy API, continue to use the same `gwcc 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, ClaimCenter generates the libraries and Javadoc for the deprecated API in a slightly different location. See “Java Library Generation in Version 8.0” on page 134.
- 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 ClaimCenter 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 “Deploy Non-OSGi Java Classes and JARs” on page 647 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 ClaimCenter. ClaimCenter 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 128 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 135.
  - For more about plugin directories, see “Adding an Implementation to a Plugins Registry Item” on page 128 in the *Configuration Guide*.

## Overview of Java API Changes in ClaimCenter 8.0

There are many differences between the deprecated Java API and the ClaimCenter 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 “Java IDE Inspections that Flag Unsupported Internal APIs” on page 640 in the *Integration Guide*.

The packages for ClaimCenter entity types appear differently in Java than in Gosu. See “Entity Packages and Customer Extensions from Java in Version 8.0” on page 133.

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 ClaimCenter performed a shallow clone of collections that were function arguments or return values. In more recent versions of ClaimCenter, 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 135.
- 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 135.
- 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 135.
- 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 135.
- In previous releases, exceptions that ClaimCenter 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 133
- “Java Library Generation in Version 8.0” on page 134

- “Getting a Reference to a Bundle in Java” on page 134
- “Creating Entities in the ClaimCenter 8.0 Java API” on page 134
- “Calling Gosu Classes from Java” on page 135
- “Gosu Enhancements on Java Classes or Gosu Classes” on page 135

## 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, ClaimCenter 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 ClaimCenter Studio includes a code inspection you can use. See “Java IDE Inspections that Flag Unsupported Internal APIs” on page 640 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 “Access Entity and Typecode Data in Java” on page 641 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 ClaimCenter Java API, you can reference a type directly by its fully-qualified name. However, for ClaimCenter 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, ClaimCenter 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:

- “Access Entity and Typecode Data in Java” on page 641 in the *Integration Guide*
- “Access Entity Instances from Java” on page 643 in the *Integration Guide*

- “Access Typecodes from Java” on page 643 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.

### ClaimCenter 8.0 Java API

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

```
gwcc regen-java-api
```

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

```
ClaimCenter/java-api/lib  
ClaimCenter/java-api/doc
```

See “Regenerate Java API Libraries” on page 642 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:

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

ClaimCenter 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:

```
ClaimCenter/java-api/deprecated/lib  
ClaimCenter/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 ClaimCenter with the deprecated API. For a complete reference, see “Where to Deploy Your Java Classes and Libraries” on page 135. For more about plugin directories, see “Adding an Implementation to a Plugins Registry Item” on page 128 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 335 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 ClaimCenter 8.0 Java API, use the following code:

```
gw.pl.persistence.core.Bundle b = gw.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 644 in the *Integration Guide*.

## Creating Entities in the ClaimCenter 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 ClaimCenter 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 “Create New Entity Instances from Java” on page 644 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 “Access Gosu Classes from Java Using Reflection” on page 645 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 646 in the *Integration Guide*.

## Exception Handling Now Straightforward

With the legacy Java API, exceptions that ClaimCenter 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 “Access Typecodes from Java” on page 643 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 127 in the *Gosu Reference Guide*.

For minor differences in the Gosu and Java APIs for querying, see “Query for Entity Data in Java” on page 645 in the *Integration Guide*.

## Where to Deploy Your Java Classes and Libraries

### Implementing Java Plugins

If you are writing Java classes that implement ClaimCenter 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 128 in the *Configuration Guide*.

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

- Put classes in ClaimCenter/modules/configuration/plugins/*PLUGINDIR\_OR\_SHARED*/classes
- Put libraries in ClaimCenter/modules/configuration/plugins/*PLUGINDIR\_OR\_SHARED*/lib

From the previous two locations, ClaimCenter 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 ClaimCenter/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 ClaimCenter/modules/configuration/plugins/shared/basic/classes
- Put libraries in ClaimCenter/modules/configuration/plugins/shared/basic/lib

From the previous two locations, ClaimCenter 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 ClaimCenter 8.0 Java API, use the following locations:

- Put classes in ClaimCenter/modules/configuration/plugins/*PLUGINDIR\_OR\_SHARED*/basic/classes
- Put libraries in ClaimCenter/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 ClaimCenter/modules/configuration/plugins/*GOSU\_OR\_SHARED*/classes
- Put libraries in ClaimCenter/modules/configuration/plugins/*GOSU\_OR\_SHARED*/lib

From the previous two locations, ClaimCenter 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 ClaimCenter/modules/configuration/plugins/Gosu/classes
- Put libraries in ClaimCenter/modules/configuration/plugins/Gosu/lib

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

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

From the previous two locations, ClaimCenter 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 ClaimCenter 8.0 Java API, use the following locations:

- Put classes in `ClaimCenter/modules/configuration/plugins/GOSU_OR_SHARED/basic/classes`
- Put libraries in `ClaimCenter/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 ClaimCenter Studio, the user interface changed for registering plugins. 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.

ClaimCenter uses the location of classes and libraries on disk to determine whether to use the legacy Java class loader or the ClaimCenter 8.0 Java class loader.

### See Also

- “Using the Plugins Registry Editor” on page 127 in the *Configuration Guide*.
- “Where to Deploy Your Java Classes and Libraries” on page 135.
- For the 8.0 Java API, see “Deploy Non-OSGi Java Classes and JARs” on page 647 in the *Integration Guide*.

## Document Management Integration Changes

This topic includes:

- “Document Assistant is a Java Applet” on page 137
- “Client-side PDF Document Production Removed” on page 137
- “Document Content Response Types Removed: HTML and JSCRIPT” on page 138
- “Configuration Parameter Changes” on page 138
- “Minor Change in Document Management Mailmerge Attribute” on page 138

### 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.

### 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, ClaimCenter 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.

## ClaimCenter Contact Plugins Changes

The `IAddressBookAdapter` and `IContactSearchAdapter` plugin interfaces are deprecated. Although the old plugin interfaces continue to work in this release, immediately begin to transition to the `ContactSystemPlugin` plugin interface as a replacement.

It is unsupported to implement both the deprecated `IContactSearchAdapter` plugin and the new `ContactSystemPlugin` plugin interface at the same time. It is unsupported to implement both the deprecated `IAddressBookAdapter` plugin and the new `ContactSystemPlugin` plugin interface at the same time.

Also, ClaimCenter includes an implementation of a `MessageRequest` plugin to handle late-bound address book UID values in contact-related messages.

### See also

- For more information about the plugin changes and the new messaging flow for contacts, see “Integrating with a Contact Management System” on page 495 in the *Integration Guide*.
- For related contact management changes, see “New And Changed Features in ContactManager 8.0.0” on page 20 in the *Contact Management Guide*.

## PolicyCenter Product Model Import into ClaimCenter Changes

PolicyCenter 8.0 removes the Product Model editor from Studio and replaces it with a separate web application included with PolicyCenter 8.0. The web application, Product Designer, does not include a menu option to run the ClaimCenter Typelist Generator. Instead, you run the ClaimCenter Typelist Generator from the command line.

### See also

- “Running the ClaimCenter Typelist Generator” on page 524 in the *Integration Guide*.



---

## part IV

# Release Notes Archive

This section contains the release notes for previous versions of ClaimCenter. Use these files to learn what features changed from one release to another.

---

**IMPORTANT** This section contains upgrade information originally provided for earlier ClaimCenter releases. It may be superseded by later release notes or other upgrade documentation.

---



---

## chapter 19

# Guidewire ClaimCenter 8.0.4 Release Notes

## Release 8.0.4

### 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:** 23-June-2016

Copyright © 2001-2016 Guidewire Software, Inc.

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, Guidewire Spotlight, Gosu, Adapt and succeed, 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.

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.

# 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 ClaimCenter.

---

**IMPORTANT** If you skipped one or more upgrade releases to ClaimCenter, be sure to read the release notes for those releases to learn about changes and fixed issues.

---

## Version Number

This release of Guidewire ClaimCenter is 8.0.4.

- If you are integrating with Guidewire PolicyCenter, this version of ClaimCenter requires PolicyCenter 7.0.0 or later. The latest maintenance release is preferred.
- If you are integrating with Guidewire ContactManager, this version of ClaimCenter requires ContactManager 8.0.4 or later. The latest maintenance release is preferred.

## Installing This Release

Refer to the following:

- The *ClaimCenter Installation Guide* for general installation information
- Prior ClaimCenter release notes for any versions that you have skipped

## Upgrade Information

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 ClaimCenter 8 release to another ClaimCenter 8 release, running the Configuration Upgrade Tool may produce following warning message:

```
WARN cannot find Emerald base configuration zip, this could indicate a problem...
```

If the `ClaimCenter/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 section contains issues and major changes that may affect your installation. For information on new features and major changes, see the topic “New and Changed in ClaimCenter 8.0.4” in the ClaimCenter New and Changed Guide.

- Base PCF File Changes
- Base Rule Changes
- Changes in This Release Provided in Upgrade Diff Report

### 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.

#### **ClaimCenter release 8.0.3 to 8.0.4**

To view a report of the changes to the base PCF files, *[click here](#)*.

### Base Rule Changes

#### **ClaimCenter release 8.0.3 to 8.0.4**

There are no changes to the base rules.

### 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 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

### Application Improvements and Resolved Issues

ID	Description
<b>Activities, Activity Patterns, Workplan</b>	

CLM-25100	Resolved an issue with incorrect activity pattern creation when a check is rejected. The system may create an activity with a pending_create_rejected or pending_update_rejected pattern instead of approval_denied. Only the approval_denied pattern should have its Type column set to approvaldenied. The other two patterns were updated in system_data.xml and activity-patterns.xml to be of type general.
-----------	--

**Assignment, Email, Notes, Documents, Question Sets**

CLM-31250	The template_tool now allows for validating email and note templates as well as document templates. There are new, additional, options to list and validate document, email, and note templates. The new options are: <ul style="list-style-type: none"> <li>-list_doc_templates</li> <li>-list_note_templates</li> <li>-list_email_templates</li> <li>-list_all_templates (Runs all three -list cases above in sequence.)</li> <li>-validate_all_doc_templates</li> <li>-validate_all_note_templates</li> <li>-validate_all_email_templates</li> <li>-validate_all_templates (Runs all three -validate cases above in sequence.)</li> <li>-validate_doc_template ID</li> <li>-validate_note_template ID</li> <li>-validate_email_template ID</li> </ul>
-----------	---

**Claim File - Associations, Calendar, Claim Status, Evaluation, Exposures, History, Incidents, Loss Details, Matters, Negotiation, SIU, Summary**

CLM-28094	Fixed a refresh issue in the Loss Details page that prevented the New Contact menu from appearing on the Witnesses ListView.
-----------	--

**Claim Metrics, Performance**

CLM-30748	Resolved performance degradation issue in screens that trigger claim and exposure metrics calculation.
-----------	--

**Command-line Tools, Toolkit, Data Dictionary, External Entities, CC Batch Process, Performance**

CLM-30586	Bulk Purge batch processing has now been implemented as a work queue.
-----------	---

**Contacts, Roles**

CLM-29574	Fixed an issue where creating a check with a payee attached to other checks could cause submitted transactions to be revalidated.
CLM-29689	Added a new version of ContactSystemUtil#importContactFromContactSystem that takes a bundle, allowing the caller to specify the bundle the imported Contact should be placed in.
CLM-30630	Fixed an issue where the creation of a primary contact at the same time as a vendor contact caused the primary contact to be created in ContactManager without requiring approval.

**Financials - Checks, Payments, Deductibles**

CLM-29816	Fixed an issue where retired checks that are still assigned to a check group caused the FinancialsCalculations batch process to fail.
CLM-30460	Fixed an issue where the TransferredCheck property on Check was returning itself if the check was transferred and had a recoded payment. Now, it returns null. See the Javadoc for more information on the TransferredCheck and TransferredToCheck properties.

**I18N, Language Pack, Localized Modules**

CLM-29720	Fixed an issue where an error was shown on the Vehicle Incident page when Rental Daily Rate was used.
-----------	---

**Infrastructure, possible Platform**

CLM-30396	Fixed an issue where the HTML in RangeCells was not escaping properly.
-----------	--

**Integration ISO**

CLM-29572	Fixed an issue where ISOReplyBase could throw a NullPointerException if a response from ISO was formatted with MatchDetails missing in a specific sequence.
-----------	---

**Policy, LOB config**

CLM-29576	Added Retired to the policyvehicle1 unique index for the VehicleRU entity.
-----------	--

**Policy Refresh**

CLM-30083	Fixed graph traversal in features such as Policy Refresh so that traversing large graphs does not cause a StackOverflowError. Previous versions had the workaround of being able to increase the stack size by using the -Xss Java option.
-----------	--

**SOAP APIs, Plugins, Rules - Core**

CLM-28918	Updated web services to make DTOs consistent with entities. <ul style="list-style-type: none"> <li>For any new implementations or modifications/upgrades to existing web services, please use the ClaimCenter 804 web services.</li> <li>Maintenance of DTOs should be done by manual editing. The DTO generator should now only be used to generate a new DTO.</li> </ul>
-----------	--

## Platform Improvements and Resolved Issues

The following are the primary improvements and issues corrected in this release:

ID	Description
<b>Appserver Support - WebSphere</b>	
PL-31805	EAR built for WebSphere now contains an IBM-specific descriptor that sets classloading mode to parent-last and WAR classloader policy to single. Setting these parameters in WebSphere manually was required for prior releases. Verify that classloading mode and WAR classloader policy are set correctly after the installation.
<b>Configuration Upgrade</b>	
PL-32082	The configuration upgrade tool would report fatal errors with case issues in display keys. The tool now corrects these issues and proceeds with the configuration upgrade.
<b>Database Instrumentation</b>	
PL-25165	Guidewire has modified the Database Table Info download to include the XML database configuration files. To access the files, download the table information file from the Server Tools → Database Table Info screen, then open the index.html file, and then click config_files: Directory with config files.
PL-31967	Added two new options to the table_import command to list and download load history reports.
<b>Database Support - Oracle</b>	
PL-31514	LOB objects associated with spatial columns are now stored in a LOB tablespace, if so configured for the table.
<b>Database Support - SQL Server</b>	
PL-31729	LOB objects associated with spatial columns are now stored in a LOB tablespace, if so configured for the table.
PL-32898, CLM-31239	Guidewire has improved the performance of the production of the (Server Tools Info Pages) SQL Server DMV Snapshot by restricting the data extract to the top 400 queries.
<b>Database Upgrade</b>	
PL-32115	A SQL Server upgrade that runs only the version trigger that converts integers to big integers for ID and foreign keys now runs without error.
<b>Document Management</b>	
PL-32605	Resolved an issue that caused a ClassCastException exception in document generation if you set IDocumentTemplateSource.gwp plugin parameter cacheDescriptors to true.
<b>Gosu</b>	
PL-30881	Resolved an issue with GosuProgram threads retaining type system locks unnecessarily, which resulted in deadlock.
PL-32490	Fixed an issue where Gosu does not automatically downcast if the left side of the typeis or typeof expression uses deprecated members.

ID	Description
<b>IntelliJ IDE - Typelist Editor</b>	
PL-30524	Fixed an issue where Studio would encode files containing non-English characters as ANSI.
<b>Messaging</b>	
PL-31680	Resolved a thread safety issue that occurred in rare cases when initializing event messages.
<b>Other</b>	
PLWEB-161	Fixed an issue where printing list views would throw an exception.
PLWEB-1997	Fixed an issue where an ampersand or other special character would break auto-filling other data.
PLWEB-2032	Added a width attribute to ToolbarFilter to specify a width in pixels. This allows you to increase the width of a ToolbarFilter if the default width does not allow all filter options to be displayed
PLWEB-4568	Restored the ability to hide the sidebar.
PLWEB-5	Fixed an issue where the progress bar could be triggered multiple times and cause an incorrect status to be shown.
<b>Other - Cloud</b>	
PL-27758	CSV export from a list view now correctly handles newline characters (CR, LR) in the source data.
PL-32043	Upgraded the Joda-Time library to version 2.5 for updated time zone information.
<b>Other - Integration Pod</b>	
PL-31798	The utility class instance for the production data change utility methods changed packages. If you use the access syntax that the documentation recommends (DataChange.util), your code requires no changes.
<b>PCF - Layout - List View</b>	
PLWEB-2017	In a RowTree, a Row widget that has its highlighted property set to true now displays as highlighted.
<b>Staging Tables</b>	
PL-30911	(Oracle) Guidewire has added a <loader> element to database-config.xml that you can use to configure parallel operations during staging table load.
PL-31439	ClaimCenter provides a configuration to run staging table based data loading with parallelism on Oracle.
PL-31931	If an integrity check and load fails after the check, another load operation on the same database is not performed.
<b>Web - Configuration</b>	
PLWEB-2000	Fixed an issue where the regional calendar for Portuguese displayed extraneous characters.
PLWEB-2035	Resolved an issue where the combination date-time widget would not auto-fill with a standard time when only the date was entered, if the date contained leading zeroes in the month or day.
PLWEB-4251	Added a visible attribute to the TabBarUnsavedWork and the MenuActions button to allow dynamic enabling and disabling of their visibility based on Gosu logic.
<b>Web - IE Support</b>	
PLWEB-18	As a security precaution, added an explicit header to prevent MIME sniffing.
<b>Web - Styling</b>	
PLWEB-1993	Fixed an issue where dividers were not appearing in the Actions menu.
PLWEB-4289	Fixed an issue where the labelStyleClass was not being applied properly to inputs in the GlobalAddressInputSet.
<b>Web - UI/Runtime</b>	
PLWEB-140	Fixed an alignment issue with check boxes in input groups.
PLWEB-15	Fixed an issue where dates with leading zeroes in the month and year would not autoformat correctly for all localization options.
PLWEB-1728	Improved keyboard navigation of the Actions menu.
PLWEB-1918	Fixed an issue where sorting a tree table would throw an exception.
PLWEB-2003	Fixed an issue where an unsecured GET request could be exposed.

ID	Description
PLWEB-2005	Fixed an issue where pressing Enter on a check box would both toggle the check box and submit the page.
PLWEB-2010	Fixed an issue where numCols did not appropriately scale for the font size. For example, year fields specified with numCols = 4 did not show all four digits. This may affect the layout of screens where numCols has previously been set.
PLWEB-2034	Fixed an issue with the target property of the ExitPoint widget that prevented it from functioning properly.
PLWEB-2039	Pressing Enter no longer changes the value of a check box. Now only pressing Space changes it.
PLWEB-2040	Fixed an issue where a button with the download attribute set to true lost the chosen values from the multiselect shuttle widget.
PLWEB-2047	Fixed an issue where help text would not appear for boolean radio button inputs.
PLWEB-2079	Fixed the info bar to allow for better copying and pasting of individual text snippets.
PLWEB-2191	Modified info bar styling to better reflect which items are links instead of text.
PLWEB-3042	Modified styling of input groups to better delineate expansion and separation.
PLWEB-32	Fixed an issue where inputs would not reset to the default value after enabling and disabling.
PLWEB-37	A date and time entered without separators, such as "11272014 1035am", is now accurately translated and validated. Note that a space must separate the date from the time.
PLWEB-4199	Fixed an issue where the objFocusID was not escaping HTML properly.
PLWEB-4521	Fixed an issue where the HTML in a RangeCell was not escaping properly.
PLWEB-47	Fixed an issue where ClaimCenter would hang in Internet Explorer when creating new services.
<b>Web Services - WSI (New)</b>	
PL-32548	Fixed a security vulnerability to XML external entity (XXE) attacks. For more information about this issue, or to learn about temporary workarounds while you deploy this fix, see Knowledge Base article 3078 at <a href="https://guidewire.custhelp.com/app/answers/detail/a_id/3078/kw/3078">https://guidewire.custhelp.com/app/answers/detail/a_id/3078/kw/3078</a> .
<b>XMLNode</b>	
PL-31602	Resolved an XML parsing issue due to the compatibility-xsd.xml file omitting an entry for gw.xml.xsd.xsdtypes.

## Known Issues and Limitations

This section describes known issues with this release of Guidewire ClaimCenter:

- “ClaimCenter Known Issues” on page 149
- “Studio/Platform Known Issues” on page 151

**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 sections. The goal of this policy is to make upgrades as straightforward as possible.

### ClaimCenter Known Issues

#### Upgrade trigger on CCAssigneeInput widget is either missing or broken (CLM-26109)

**Issue** – The upgrade trigger for removing showNoneSelected="false" on CCAssigneeInput widget is either missing or broken.

**Workaround** – In the CCAssigneeInput widget, replace usage of the showNoneSelected attribute with the required attribute set to true or a default value.

## Too many compile warnings in ClaimCenter Studio (CLM-26978)

**Issue** – The number of warnings when compiling code in ClaimCenter Studio make it hard to see actual problems.

**Workaround** – Deprecation of RPC web services causes the largest number of warnings. If you are not using RPC web services, you can safely ignore warnings regarding `RpcWebService` or classes with `soap` in their package names.

A smaller number of warnings are related to the section inclusion limit for second pass verification of PCF files. The default limit is 1,000 for performance reasons, but a number of PCF files are more complicated than that. If you want to get rid of these warnings, you can change the setting manually in Guidewire Studio by navigating to **File → Settings → Guidewire Studio**. In the **Guidewire Studio** page, change the settings for **PCF Settings**. You can turn off **Limit Second Pass Verification** altogether or set **Section Inclusion Limit** to 3000, which is high enough for all the PCF files in the base configuration of ClaimCenter.

There is no workaround for the rest of these warnings.

## FNOL Snapshot cannot support upgraded property claims (CLM-26997)

**Issue** – In the ClaimCenter 8.0.1 **FNOL Snapshot → Loss Details** PCFs, there is a reference to a property that did not exist in ClaimCenter 8.0.0. As a result, FNOL snapshots created for property claims in 8.0.0 cannot be viewed in 8.0.1.

**Workaround** – The workaround necessitates some changes in your PCF files. Edit `ClaimSnapshotGeneralPR800PanelSet.default.pcf`, `ClaimSnapshotGeneral1800PanelSet.auto.pcf`, `ClaimSnapshotGeneral1800PanelSet.g1.pcf`, and `ClaimSnapshotGeneral1800PanelSet.Trav.pcf`, as follows:

- Replace `FixedPropertyIncidentInstance.getProperty().Address.DisplayAddressLine1` with  
`(FixedPropertyIncidentInstance.getProperty().Address.AddressLine1Kanji as String).HasContent ? FixedPropertyIncidentInstance.getProperty().Address.AddressLine1Kanji : FixedPropertyIncidentInstance.getProperty().Address.AddressLine1.`
- Replace `FixedPropertyIncidentInstance.getProperty().Address.DisplayCity` with  
`(FixedPropertyIncidentInstance.getProperty().Address.CityKanji as String).HasContent ? FixedPropertyIncidentInstance.getProperty().Address.CityKanji : FixedPropertyIncidentInstance.getProperty().Address.City.`
- Replace `prop.Property.Address.DisplayAddressLine1` with  
`(prop.Property.Address.AddressLine1Kanji as String).HasContent ? prop.Property.Address.AddressLine1Kanji : prop.Property.Address.AddressLine1.`
- Replace `prop.Property.Address.DisplayCity` with `(prop.Property.Address.CityKanji as String).HasContent ? prop.Property.Address.CityKanji : prop.Property.Address.City.`

## Date in Search by Contact results unclear (CLM-27001)

**Issue** – When you search for claim contacts using **Search by Contact**, the **Search Results** pane displays a date for search results that does not map to the Loss Date for the claim, as expected.

**Workaround** – None. Guidewire is aware of this issue.

## Extending ActivityPattern results in an exception during upgrade (CLM-27857)

**Issue** – The `AddActivityPatternsFromImportDataVersionTrigger` upgrade trigger throws an exception if the `ActivityPattern` entity is extended.

Two scenarios can cause this exception:

- Case 1 – Adding a new column to `ActivityPattern` during the upgrade process.
- Case 2 – Adding a non-nullable column (`nullOk = false`) to `ActivityPattern` during the upgrade process.

**Workaround** – The following steps can be taken, depending on the case.

- Case 1 – Add the column to the previous version of ClaimCenter before performing the major version upgrade. If this is not possible, add the column after the upgrade is complete.
- Case 2 – The column to be added must be nullable (`nullOk = true`), and ClaimCenter must alter the column before the major version upgrade is attempted. You can change it back after the upgrade.

### Assertion error when clicking Unsaved Work from Close Claim (CLM-28002)

**Issue** – An `AssertionError` is generated when you select **Actions** → **Close Claim** and click **Unsaved Work** → **Claim Number**.

**Workaround** – Use the following steps:

1. In Guidewire Studio, open `CloseClaimPopup.pcf`.
2. Click on the PCF to display its properties.
3. In the **Properties** window, change the value of the `countsAsWork` attribute to `false`.

This ensures that changes to the `CloseClaimPopup` PCF are not evaluated as work and consequently, the **Unsaved Work** icon remains unavailable.

## 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, ClaimCenter throws `ParseResultsException`. This is the intended behavior.

**Workaround** – Restart the workflow engine. To do so:

1. Log in to ClaimCenter 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 `@WsiPermission` annotation.

**Workaround** – Do the following:

From the `ClaimCenter/bin` directory, at a command prompt type the command:

```
gwcc regen-soap-api
```

In Windows Explorer, copy the WSDL files from the location:

```
ClaimCenter/soap-api/wsi/wsdl
```

to:

ClaimCenter/admin/res/ws!

### 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 20

# Guidewire ClaimCenter 8.0.3 Release Notes

## Release 8.0.3

### 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:** 23-June-2016

Copyright © 2001-2016 Guidewire Software, Inc.

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, Guidewire Spotlight, Gosu, Adapt and succeed, 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.

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.

# 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 ClaimCenter.

---

**IMPORTANT** If you skipped one or more upgrade releases to ClaimCenter, be sure to read the release notes for those releases to learn about changes and fixed issues.

---

## Version Number

This release of Guidewire ClaimCenter is 8.0.3.

- If you are integrating with Guidewire PolicyCenter, this version of ClaimCenter requires PolicyCenter 7.0.0 or later. The latest maintenance release is preferred.
- If you are integrating with Guidewire ContactManager, this version of ClaimCenter requires ContactManager 8.0.3 or later. The latest maintenance release is preferred.

## Installing This Release

Refer to the following:

- The *ClaimCenter Installation Guide* for general installation information
- Prior ClaimCenter release notes for any versions that you have skipped

## Upgrade Information

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 ClaimCenter 8 release to another ClaimCenter 8 release, running the Configuration Upgrade Tool may produce following warning message:

WARN cannot find Emerald base configuration zip, this could indicate a problem...

If the `ClaimCenter/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 section contains issues and major changes that may affect your installation. For information on new features and major changes, see the topic “New and Changed in ClaimCenter 8.0.3” in the ClaimCenter New and Changed Guide.

- Base PCF File Changes
- Base Rule Changes

- Changes in This Release Provided in Upgrade Diff Report

**IMPORTANT** There is an issue with the usage of `cacheDescriptors` on the `IDocumentTemplateSource` plugin. Guidewire recommends that caching of document templates and descriptors be disabled in production systems. See Knowledge Base article 2973 for more information.

## 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.

### ClaimCenter release 8.0.2 to 8.0.3

To view a report of the changes to the base PCF files, [click here](#).

## Base Rule 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.

### ClaimCenter release 8.0.2 to 8.0.3

To view a report of the changes to the base rules, [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 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
<b>Claim Metrics, Performance</b>	
CLM-27036	Improved the performance of claim financial metrics by removing the use of <code>viewEntity</code> data objects and loading entities directly instead for the calculations.
<b>Financials - ReserveLines, TAccounts, Financials Calculations</b>	
CLM-27900	Added a way for the <code>getAmount</code> enhancement methods on <code>FinancialsCalculations</code> to return zero when there is no transaction that matches the calculator's criteria. See <code>GWFinancialsCalculationGetAmountEnhancement</code> and the <code>ForceFinancialsCalculationGetAmountMethodsNotReturnNull</code> configuration parameter.
CLM-28558	When new foreign exchange adjustments were applied to payments that already had adjustments, the reserving balances were not being updated accurately. This is now fixed. This may affect customers on Emerald who use multicurrency and foreign exchange adjustments.
<b>Financials - SOAP APIs, Plugins, Rules</b>	
CLM-27886	Deprecated web services, plugins, and classes for integration with ContactManager 7. ClaimCenter 8 must be integrated with ContactManager 8.
<b>Infrastructure, possible Platform</b>	
CLM-28768	Remnants of the <code>ClaimPickerInput</code> widget have now been deleted. Use <code>PickerInput</code> instead.

CLM-29066	This patch corrects a performance regression issue with document management. [UPDATED: This issue description has been updated from a previous version of these Release Notes. Please see the important note in "Major Issues and Changes" on page 155.]
<b>Integration CM</b>	
CLM-26159	In prior releases of ClaimCenter, every use of the Address Book Search screen enabled the user to launch ContactManager. However, launching ContactManager from a worksheet did not make sense, so now it is not possible to launch ContactManager from a worksheet. You can still launch ContactManager from the Address Book Search screen itself.
CLM-28429	An issue where ContactTag elements were erroneously included in a PendingUpdate has been resolved.
CLM-28430	Subsequent updates to elements associated with a contact, such as array elements and foreign keys, were causing additional entities to be erroneously created in ContactManager. This issue has now been fixed.
CLM-28754	Previously, the web browser exit point from ClaimCenter to ContactManager used the URL defined in suite-config.xml, designed for ContactManager suite integration. For example, the <b>Edit in ContactManager</b> button, available in some ClaimCenter contact editing screens, uses this URL to open ContactManager in another web browser window. If you want to provide separation between web browser and application integration URL definitions, you can now define this exit point in config.xml in the ContactSystemURL configuration parameter.
<b>Integration CM, Integration CM - Core: Link, Related Contacts, Match, Retrieve, Search, Update</b>	
CLM-28371	An issue where the syncable properties were being computed too many times for contact fingerprint and syncing has been fixed.
CLM-28878	An issue where erroneous updates to ContactManager were generated for array elements on contacts has been fixed.
<b>Integration ISO</b>	
CLM-28242	Updated ISO's request and response XSD files to the latest version and made necessary changes to ISOReplyBase to support the changed structure while minimally affecting custom code.
<b>Performance</b>	
CLM-27924	Poorly performing queries for entities that caused large entity loads into memory have been fixed.
CLM-28072	Improved claim search performance by adding person name indexes in ClaimContact.etc. The added indexes are disabled in the base configuration and can be optionally enabled and partitioned using database-config.xml. The original index, PersonLastFirstN, was renamed to ClmConLastFirstN and moved to ClaimContact.etc for better configurability.
CLM-28119	Added new AddIndexHintForLossDate configuration parameter that adds a hint to the index Claim.claimu7u to improve the Oracle performance of the <b>Desktop Activities</b> screen.
CLM-28217	An Oracle query hint was added to disable index fast full scans during desktop activities queries. A configuration parameter, DisableIndexFastFullScanForDesktopActivities, was added to disable this hint, should it prove detrimental in certain circumstances.
CLM-28235	Improved performance of PercentEscalatedActivitiesExposureMetricMethodsImpl for claims with large quantities of exposures by only updating metrics if relevant activities are changed in the bundle. Previously, metrics would be updated if any activities were changed in the bundle.  Additionally, an Activity array was added on Exposure to avoid indirectly loading activities from a claim, where possible.
<b>Printing</b>	
CLM-28849	An issue where printing policy details for a claim, including FNOL data, caused an exception has been resolved.

## Platform Improvements and Resolved Issues

The following are the primary improvements and issues corrected in this release:

ID	Description
<b>Authentication</b>	
PL-30775	Removed the unused configuration parameter <code>ShouldSyncUserRolesInLDAP</code> .
<b>Batch Processes</b>	
PL-30718	The free-text batch load command now works correctly with JNDI data sources.
<b>Build Infrastructure</b>	
PL-27679	The <code>gwcc regen-java-api</code> command now generates Java API Javadoc into the <code>java-api/doc/api</code> directory.
<b>Command Line Tools</b>	
PL-27608	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> . Also added new option <code>-e</code> , which enables you to specify a semicolon-separated list of directories to exclude.
<b>Consistency Checker</b>	
PL-30098	Fixed an issue where Database Consistency Checks that checked date parameters would compare against an incorrect date.
<b>Database Support - DB2</b>	
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 true 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 false in the base configuration.
<b>Database Support - Oracle</b>	
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 a bug that mistakenly set Oracle LOBs 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 <code>Errors</code> 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.
<b>Database Support - SQL Server</b>	
PL-30578	Two pages have been added to the SQL Server DMV performance report that aggregate active session statistics by user and by action.
<b>Document Management</b>	
PL-31555	Fixed a bug so you can now use asynchronous document storage ( <code>AsyncDocumentContentSource</code> ) even if the Plugins registry has an enabled implementation for the <code>IDocumentMetadataSource</code> plugin interface.
<b>Email</b>	

ID	Description
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.
<b>Entities/Metadata</b>	
PL-30357	Fixed an issue where the <code>gwcc 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.
<b>External Entities</b>	
PL-28992	Resolved an issue that caused the <code>gwcc regen-java-api</code> command to sometimes fail if the flag <code>-Ddeprecated=true</code> was specified.
<b>Geocoding/Proximity Search</b>	
PL-28585	Driving directions are now in the user's preferred language instead of the language of the locale for the user's preferred regional formats. For example, if the user's preferred language is French and preferred regional formats are Japan, driving instructions are in French.
<b>Gosu</b>	
PL-29715	Fixed a bug that could cause runtime exceptions if you set the <code>config.xml</code> parameter <code>WarnOnImplicitCoercion</code> to the value <code>false</code> .
<b>IntelliJ IDE - Compiler</b>	
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.
<b>IntelliJ IDE - Entity Editor</b>	
PL-29124	Fixed an issue where comment elements such as <code>&lt;!-- Comment --&gt;</code> inside an entity or typelist definition file were removed when edited in Studio.
<b>IntelliJ IDE - Localization</b>	
PL-30209	You can now switch the language in Studio without needing to restart. To do so, navigate to <b>File → Settings → Guidewire Studio</b> , and then under <b>Language Settings</b> , click the language.
<b>IntelliJ IDE - PCF Editor</b>	
PL-30102	Removed a non-functional link in the pop-up help text when hovering over a PCF widget in Studio.
<b>IntelliJ IDE - Plugins</b>	
PL-29776	If you configure Guidewire Studio to work with IDEA IntelliJ Ultimate, now the OSGi Plugin Editor also uses Ultimate.
<b>IntelliJ IDE - Product Model Editor</b>	
PL-29996	Fixed an issue that prevented creating Gosu enhancements for product model coverages.
<b>IntelliJ IDE - Webservices Editor</b>	
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.
<b>Messaging</b>	
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, ClaimCenter 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"> <li>The <code>InboundIntegrationPlugin</code> plugin interface is now called <code>InboundIntegrationStartablePlugin</code></li> <li>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</li> <li>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.</li> </ul>
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.
<b>Other - Developer IDE Pod</b>	
PL-27489	Reduced the number of unnecessary library files that <code>regen-java-api</code> generates into the <code>lib</code> folder.
<b>Other - Integration Pod</b>	
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>&lt;custom-integration&gt;</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.
<b>Other - Persistence Pod</b>	
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.
<b>Other - Web UI Pod</b>	
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"> <li>/ (slash; example: 18/2/2015)</li> <li>. (dot; example: 18.2.2015)</li> <li>- (dash; example: 18-2-2015)</li> <li>&lt;space&gt; (example: 18 02 2015)</li> <li>&lt;nospace&gt; (example: 18022015)</li> </ul>
PL-31050	Fixed an issue where the <code>DateTime</code> input widget was not navigable with the keyboard.
<b>Queries</b>	
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
<b>Search</b>	
PL-29222	<p>This release introduces the env attribute to the &lt;document&gt; element in solrserver-config.xml. You can use the value you set in solrserver-config.xml as a Java VM environment variable to run ClaimCenter with that Guidewire Solr Extension instance without editing the solrserver-config.xml. For example, your solrserver-config.xml file contains the following document elements.</p> <pre>&lt;PC-only content&gt; &lt;document name="policy" archive="false" servername="embedded"/&gt; &lt;document name="policy" archive="false" servername="localhttp" env="local"/&gt; &lt;/PC-only content&gt; &lt;CC-only content&gt; &lt;document name="claimcontact" archive="true" servername="embedded"/&gt; &lt;document name="claimcontact" archive="true" servername="localhttp" env="local"/&gt; &lt;/CC-only content&gt;</pre> <p>By default, ClaimCenter runs with the Solr server found in the first document element in solrserver-config.xml. In the preceding example, ClaimCenter runs with the embedded server by default. If you stop the ClaimCenter application server and restart it with -Dgw.cc.env=sq1 as an option, ClaimCenter runs with the Solr server found in the document element with the attribute env="sq1". In the preceding example, ClaimCenter runs with an external instance of the Guidewire Solr Extension hosted locally.</p>
PL-29476	Added the zkhosts property to Guidewire Solr Extension server configuration to specify in a comma separated list the hosts and ports of all members of the Zookeeper ensemble. Deprecated the host and port properties, which specify only a single Zookeeper instance.
PL-29911	Upgraded the Apache Solr free text search engine to version 4.7.2.
PL-31446	Configuration of application servers changed due to changes in logging. See "Free-text Search Setup" in the <i>Installation Guide</i> for the modified setup procedures.
<b>Web - Configuration</b>	
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.
<b>Web - IE Support</b>	
PL-29681	Fixed an error where the server would report that the file toolbar-top-info-bg.gif was missing.
<b>Web - ListViews</b>	
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.

ID	Description
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.
<b>Web - Other</b>	
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.
<b>Web - Styling</b>	
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.
<b>Web - UI/Runtime</b>	
PL-28186	Fixed an issue where the PrivacyInput field would be exposed if used during postOnChange.
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 ClaimCenter 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).

ID	Description
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.
<b>Web Services - WSI (New)</b>	
PL-31055	Fixed a bug in WSDL generation for services with multiple @Throws annotations for the same exception but different reasons. The bug 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.
<b>Work Queues</b>	
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, <code>WorkItemPriorityMultiplierSecs</code> , 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 <code>findTargets</code> method, implement the <code>buildBulkInsertSelect</code> method.
<b>XMLElement (and XSD types)</b>	
PL-28345	XSD types based on XMLNode have been deprecated, and will be removed in a future release.
PL-30793	Fixed a bug in WSDL import relating to handling simple data if a complexType is a complexContent extension of a compTextType with a simpleContent extension.

## Documentation Improvements and Resolved Issues

ID	Description
DOC-2765	Added documentation on developing custom work queues. See "Custom Batch Processing" in the <i>Integration Guide</i> .
DOC-6650	Added a description of the salvage functionality in ClaimCenter. See "Working with Salvage" in the <i>Application Guide</i> .
DOC-7022	Concepts and configuration of address autofill and autocomplete are now described in the topic "Address Autocompletion and Autofill" in the <i>Globalization Guide</i> .
DOC-7508	How to localize vendor services is now documented in the topic "Localizing Vendor Services" 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> .
DOC-8873	Added documentation on viewing entity property values during debugging. See "Debugging and Testing Your Gosu Code" in the <i>Configuration Guide</i> .

DOC-9464	Added documentation on the <b>Vendor Waiting</b> state. See “Lifecycle of a Service Request” in the <i>Configuration Guide</i> .
DOC-9939	Added documentation on zeroing of reserves. See “Payments and Zeroing Reserves” in the <i>Application Guide</i> .

## Known Issues and Limitations

This section describes known issues with this release of Guidewire ClaimCenter:

- “ClaimCenter Known Issues” on page 164
- “Platform/Studio Known Issues” on page 166

**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 sections. The goal of this policy is to make upgrades as straightforward as possible.

### ClaimCenter Known Issues

#### Upgrade trigger on CCAssigneeInput widget is either missing or broken (CLM-26109)

**Issue** – The upgrade trigger for removing `showNoneSelected="false"` on CCAssigneeInput widget is either missing or broken.

**Workaround** – In the CCAssigneeInput widget, replace usage of the `showNoneSelected` attribute with the `required` attribute set to `true` or a default value.

#### Too many compile warnings in ClaimCenter Studio (CLM-26978)

**Issue** – The number of warnings when compiling code in ClaimCenter Studio make it hard to see actual problems.

**Workaround** – Deprecation of RPC web services causes the largest number of warnings. If you are not using RPC web services, you can safely ignore warnings regarding `RpcWebService` or classes with `soap` in their package names.

A smaller number of warnings are related to the section inclusion limit for second pass verification of PCF files. The default limit is 1,000 for performance reasons, but a number of PCF files are more complicated than that. If you want to get rid of these warnings, you can change the setting manually in Guidewire Studio by navigating to **File → Settings → Guidewire Studio**. In the **Guidewire Studio** page, change the settings for **PCF Settings**. You can turn off **Limit Second Pass Verification** altogether or set **Section Inclusion Limit** to 3000, which is high enough for all the PCF files in the base configuration of ClaimCenter.

There is no workaround for the rest of these warnings.

#### FNOL Snapshot cannot support upgraded property claims (CLM-26997)

**Issue** – In the ClaimCenter 8.0.1 **FNOL Snapshot → Loss Details** PCFs, there is a reference to a property that did not exist in ClaimCenter 8.0.0. As a result, FNOL snapshots created for property claims in 8.0.0 cannot be viewed in 8.0.1.

**Workaround** – The workaround necessitates some changes in your PCF files. Edit `ClaimSnapshotGeneralPR800PanelSet.default.pcf`, `ClaimSnapshotGeneral1800PanelSet.auto.pcf`, `ClaimSnapshotGeneral1800PanelSet.g1.pcf`, and `ClaimSnapshotGeneral1800PanelSet.Trav.pcf`, as follows:

- Replace `FixedPropertyIncidentInstance.getProperty().Address.DisplayAddressLine1` with  
`(FixedPropertyIncidentInstance.getProperty().Address.AddressLine1Kanji as String).HasContent ? FixedPropertyIncidentInstance.getProperty().Address.AddressLine1Kanji : FixedPropertyIncidentInstance.getProperty().Address.AddressLine1.`
- Replace `FixedPropertyIncidentInstance.getProperty().Address.DisplayCity` with  
`(FixedPropertyIncidentInstance.getProperty().Address.CityKanji as String).HasContent ? FixedPropertyIncidentInstance.getProperty().Address.CityKanji : FixedPropertyIncidentInstance.getProperty().Address.City.`
- Replace `prop.Property.Address.DisplayAddressLine1` with  
`(prop.Property.Address.AddressLine1Kanji as String).HasContent ? prop.Property.Address.AddressLine1Kanji : prop.Property.Address.AddressLine1.`
- Replace `prop.Property.Address.DisplayCity` with `(prop.Property.Address.CityKanji as String).HasContent ? prop.Property.Address.CityKanji : prop.Property.Address.City.`

### Date in Search by Contact results unclear (CLM-27001)

**Issue** – When you search for claim contacts using **Search by Contact**, the **Search Results** pane displays a date for search results that does not map to the Loss Date for the claim, as expected.

**Workaround** – None. Guidewire is aware of this issue.

### Extending ActivityPattern results in an exception during upgrade (CLM-27857)

**Issue** – The `AddActivityPatternsFromImportDataVersionTrigger` upgrade trigger throws an exception if the `ActivityPattern` entity is extended.

Two scenarios can cause this exception:

- Case 1 – Adding a new column to `ActivityPattern` during the upgrade process.
- Case 2 – Adding a non-nullable column (`nullOk = false`) to `ActivityPattern` during the upgrade process.

**Workaround** – The following steps can be taken, depending on the case.

- Case 1 – Add the column to the previous version of ClaimCenter before performing the major version upgrade. If this is not possible, add the column after the upgrade is complete.
- Case 2 – The column to be added must be nullable (`nullOk = true`), and ClaimCenter must alter the column before the major version upgrade is attempted. You can change it back after the upgrade.

### Assertion error when clicking Unsaved Work from Close Claim (CLM-28002)

**Issue** – An `AssertionError` is generated when you select **Actions** → **Close Claim** and click **Unsaved Work** → **Claim Number**.

**Workaround** – Use the following steps:

1. In Guidewire Studio, open `CloseClaimPopup.pcf`.
2. Click on the PCF to display its properties.
3. In the Properties window, change the value of the `countsAsWork` attribute to `false`.

This ensures that changes to the `CloseClaimPopup` PCF are not evaluated as work and consequently, the **Unsaved Work** icon remains unavailable.

## Platform/Studio 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, ContactManager throws `ParseResultsException`. This is the intended behavior.

**Workaround** – Restart the workflow engine. To do so:

1. Log in to ContactManager by 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 the Google Chrome browser.

**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 `@WsiPermission` annotation.

**Workaround** – Do the following:

1. From the `ClaimCenter/bin` directory, at a command prompt type the command `gwxx regen-soap-api`.
2. In Windows Explorer, copy the WSDL files from the location:  
`ClaimCenter/soap-api/wsi/wsdl`  
to  
`ClaimCenter/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 be. 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** – 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.

## DateTime widget does not auto-fill the time when the date contains leading zeroes (PL-32116)

**Issue** – The `DateTime` widget, a variation of the `DateInput` widget, configures both `dateFormat` and `timeFormat` for display. If you enter just a date into this field without entering a time, the application is supposed to auto-fill the time to 12:00 a.m. However, if the date specified contains leading zeros in both the month and day, such as 01/01/2014 for January 1, 2014, the time field is not auto-filled.

**Workaround** – A few workaround options are available for this issue:

- If users are not expected to enter the time, configure the widget to display only the date.
- Configure the display to show two separate date inputs, one for date and one for time.
- Inform users to not use leading zeros when entering the date. For example, 1/1/2014. In that case the time autofill works correctly.
- Instruct users to manually supply a time along with the date.

## Gosudoc generation fails due to Java heap settings (PL-32117)

**Issue** – The `gwcc regen-gosudoc` command fails with the following exception:

```
java.lang.OutOfMemoryError: Java heap space
```

**Workaround** – Increase the Java heap settings for the Gosudoc generation process:

1. Edit the file `ClaimCenter/modules/configuration/etc/memory.properties`

2. Add the following lines to the end of the file:

```
# Gosudoc generation memory settings  
com.guidewire.tools.gosudoc.GosuDocMain.xmx=4000
```

---

## chapter 21

# Guidewire ClaimCenter 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:** 23-June-2016

Copyright © 2001-2016 Guidewire Software, Inc.

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, Guidewire Spotlight, Gosu, Adapt and succeed, 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.

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.

## 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 ClaimCenter.

---

**IMPORTANT** If you skipped one or more upgrade releases to ClaimCenter, be sure to read the release notes for those releases to learn about changes and fixed issues.

---

### Version Number

This release of Guidewire ClaimCenter is 8.0.2.

- If you are integrating with Guidewire PolicyCenter, this version of ClaimCenter requires PolicyCenter 7.0.0 or later. The latest maintenance release is preferred.
- If you are integrating with Guidewire ContactManager, this version of ClaimCenter requires ContactManager 8.0.2 or later. The latest maintenance release is preferred.

## Installing This Release

Refer to the following:

- The *ClaimCenter Installation Guide* for general installation information
- Prior ClaimCenter release notes for any versions that you have skipped

### Upgrade Information

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 ClaimCenter 8 release to another ClaimCenter 8 release, running the Configuration Upgrade Tool may produce following warning message:

WARN cannot find Emerald base configuration zip, this could indicate a problem...

If the `ClaimCenter/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 section contains issues and major changes that may affect your installation. For information on new features and major changes, see the topic “New and Changed in ClaimCenter 8.0.2” in the ClaimCenter New and Changed Guide.

- Base PCF File Changes
- Base Rule Changes
- Changes in This Release Provided in Upgrade Diff Report

## 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.

### ClaimCenter release 8.0.0 to 8.0.2

To view a report of the changes to the base PCF files, [click here](#).

## Base Rule 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.

### ClaimCenter release 8.0.0 to 8.0.2

To view a report of the changes to the base rules, [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 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
<b>Contacts, Roles, Integration CM - Core: Link, Related Contacts, Match, Retrieve, Search, Update</b>	
CLM-25581	Previously, in the base configuration, the Adjuster role included permissions ( <code>abedit</code> , <code>anytagedit</code> ) to edit master contact records in ContactManager without manager approval. These permissions have now been removed from the Adjuster role.
<b>Desktop, Dashboard, Team, Quick Jump, Preferences, Statistics, Vacation</b>	
CLM-27119	The Open Claim Handler count under the Team tab was counting users and groups that only had Closed Claims assigned to them. This has been fixed to only count Handlers of Open Claims.
<b>DBChecks - Consistency/Integrity, Staging Tables, LoaderCallbacks, Constraints - Core, Financials - DBChecks - Consistency/Integrity, Staging Tables, LoaderCallbacks, Constraints</b>	
CLM-22054	ClaimCenter 8.0.2 now takes advantage of table groups, which provide a mechanism for running database consistency checks (DBCCs) against a named subset of tables. Table groups are configured in <code>config.xml</code> .
<b>Financials - Checks, Payments, Deductibles</b>	
CLM-13538	Fixed an issue where the amount of a new non-eroding payment was not allowed to exceed the Available Reserves on its reserve line. As non-eroding payments do not erode reserves and should not take reserves into account at all, this restriction has been removed. Similarly, <code>Payment.ExceedsAvailableReserves</code> was changed to always return <code>false</code> if the payment is non-eroding.
<b>Financials - SOAP APIs, Plugins, Rules</b>	
CLM-27128	All RPC-Encoded (RPCE) web services have been deprecated. Support for RPCE web services will be removed in 9.0. Use the WS-I versions of web services instead. Obsolete integration APIs from ClaimCenter 6.0.0 and ClaimCenter 6.0.4 were also deleted.
<b>Infrastructure, possible Platform</b>	

CLM-27209 The deferCreateArchiveIndexes attribute for database upgrade has been explicitly set to `false` in `database-config.xml`. The default value of the attribute, if omitted from the XML file, has also been set to `false` (see PL-29200).

This feature works well when this attribute is set to `true`, but its change in upgrade behavior had been unexpected. See the Upgrade Guide topic "Deferring Creation of Archive Indexes" for information about this feature.

#### Integration CM

CLM-26818 In previous releases, when a pending contact change in ClaimCenter was rejected by ContactManager, the resulting note that was created containing the requested changes was associated with the claim. This resulted in a lack of proper context for the changes in question.

With this release, the note containing the rejected changes is associated with the contact, not the claim.

#### Integration CM, Vendor Management - Services Directory, Service Requests

CLM-27016 In previous versions, in a ClaimCenter workflow, when you created a contact in ClaimCenter that did not meet the minimum requirements in ContactManager, the contact was rejected at the end of the workflow rather than at the point of contact creation. This was potentially causing confusion regarding the cause of the error, especially when there were multiple contacts, because we did not identify the offending contact.

In this release, the displayed validation error now references the specific contact that is missing required information, enabling the user to navigate back to fix the error appropriately.

#### Integration ISO

CLM-25756 In this release, ISO integration uses two separate XSD files for requests and responses. All code paths creating the request to ISO use `xsd.iso.req`, and both synchronous and asynchronous responses from ISO are parsed using `xsd.iso.resp`. If DataPower is enabled, a third XSD, `xsd.iso.ak`, is used to parse the synchronous responses.

The `handleISOReply` and `sanityCheck` methods in `ISOReplyPlugin` were changed to now accept a new `ISOReplyPayload` object instead of the XML and `HttpRequest` from ISO. The `ISOReceive` servlet places all parameters from the HTTP response into the `ISOReplyPayload` before it is passed to the `ISOReplyPlugin`.

#### Performance

CLM-27439 `ClaimID` was added to the index, `activityu0`, on the `Activity` entity in order to improve the performance of the Activity Escalation work queue.

#### Search - Claim, Activities

CLM-27854 Two utility methods, `getDistanceFromAsString` and `getDirectionsFromCache`, were added to `CCProximitySearchPageHelper` to handle null values efficiently in proximity searches. Proximity Search list views can make use of these methods instead of those in the `Address` and `DrivingDirectionsCache` classes.

#### SOAP APIs, Plugins, Rules - Core

CLM-26973 Previous versions of `importFNOLFromXML()` on `gw.api.webservice.cc.claim.ClaimAPIImpl` that were used by the RPCE `IClaimAPI` have been deprecated in favor of `importFNOLFromXMLWithoutCommit()`. The WS-I `ClaimAPI` web service now provides and commits the bundle.

CLM-26974 `ClaimAPI.bulkValidate()` has been deprecated. You should now use `ClaimAPI.startClaimValidation()`, which expects the `PublicID` of the `LoadCommand` instead of its ID. This `PublicID` is returned by the call to `TableImportAPI.integrityCheckStagingTableContentsAndLoadSourceTables()` and can be passed directly into `ClaimAPI.startClaimValidation()`.

## Platform Improvements and Resolved Issues

The following are the primary improvements and issues corrected in this release:

ID	Description
<b>Activities, Assignment</b>	
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 <code>actqueueassign</code> or <code>SystemPermissionType.TC_ACTQUEUEASSIGN</code> to the necessary role.
<b>Archiving</b>	
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.
<b>Build Infrastructure</b>	
PL-24111	When generating a WAR file using the <code>build-war</code> command and passing <code>-Dconfig.war.dictionary=true</code> , the data dictionary was not being generated. This issue is resolved.
<b>Clustering</b>	
PL-29135	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.  The JGroups watchdog uses the following algorithm: <ul style="list-style-type: none"><li>• Coordinator of the cluster sends periodic heartbeat messages. Each message contains the list of the cluster members from the coordinator's point of view.</li><li>• 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.</li></ul> This feature can be configured by adjusting two new parameters in <code>config.xml</code> : <ul style="list-style-type: none"><li>• <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.</li><li>• <code>JGroupsWatchdogMissedHeartbeatsBeforeReset</code> - The number of missed heartbeat messages after which a node resets its JGroups channel. The default is 10.</li></ul>
PL-29240	Updated the default value for <code>ClusterProtocolStack</code> to a configuration that is more reliable in case of a major network failure.  FD protocol was replaced with FD_ALL, MERGE2 was replaced with MERGE3.
<b>Configuration Upgrade</b>	
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.
<b>Consistency Checker</b>	
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.

ID	Description
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.
<b>Consistency Checker, Other - Database Pod</b>	
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.
<b>Data Distribution</b>	
PL-28620	Fixed an issue that would cause the data distribution process to fail.
<b>Database Configuration, Database Instrumentation</b>	
PL-28478	Database debug logging including SQL statements has been enabled for JNDI configurations.
<b>Database Configuration, Database Support - General, Database Support - Oracle, Database Upgrade</b>	
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.
<b>Database Configuration</b>	
PL-28660	The validationQuery attribute has been removed from the database configuration of the DBCP connection pool.
<b>Database Configuration, Database Support - Oracle</b>	
PL-28730	Updated the Database Parameters/Guidewire Database Config Statistics Settings page to show only the configured items.
<b>Database Instrumentation</b>	
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.
<b>Database Support - General</b>	
PL-28655	Improved the performance of proximity searches.
<b>Database Support - Oracle</b>	
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.
<b>Database Support - SQL Server, Database Upgrade</b>	
PL-28779	Fixed an issue that prevented upgrade to a ClaimCenter 8.0 maintenance release when the database type is SQL Server.
<b>Database Support - SQL Server</b>	
PL-28885	Fixed an issue which allowed users to create the ClaimCenter schema on a SQL Server system database.
<b>Database Upgrade</b>	
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.

ID	Description
PL-28447	If data model files were changed but the extension.properties version number was not updated, ClaimCenter 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.
<b>Entities/Metadata</b>	
PL-29932	Added the following attributes to the MonetaryAmount entity: createhistogram, createAmountHistogram, createCurrencyHistogram.
<b>Gosu</b>	
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.
<b>Gosu, IntelliJ IDE - Gosu Editor</b>	
PL-28777	Fixed an issue where Studio would display invalid "implicit conversion to integer" warnings on PCF code that used Gosu reflection.
<b>IntelliJ IDE - Customer Build</b>	
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.
<b>IntelliJ IDE - Entity Editor</b>	
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.
<b>IntelliJ IDE - Entity Editor, IntelliJ IDE - Typelist Editor</b>	
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.
<b>IntelliJ IDE - Entity Editor, IntelliJ IDE - Typelist Editor</b>	
PL-29592	Fixed an error that would occur if you tried to create an entity or typelist extension when one already exists.
<b>IntelliJ IDE - Formatting</b>	
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.
<b>IntelliJ IDE - Other</b>	
PL-27965	The Compare to Base and Show Diff windows can now be resized and maximized.

ID	Description
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 <code>studio.ultimate</code> that contains the path to the IntelliJ IDEA Ultimate installation directory.
<b>IntelliJ IDE - Typelist Editor</b>	
PL-28289	Fixed a problem when, after installing a language pack, typelists within the language pack were not recognized as valid metadata files.
<b>Internal/Server Tools Pages, Work Queues</b>	
PL-28725	Implemented the Work Queue management bean. The <b>Management Beans</b> page now shows a management bean for each work queue running locally.
<b>Localization</b>	
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: <ol style="list-style-type: none"> <li>Open Guidewire Studio and navigate in the Project window to <b>configuration</b> → <b>config</b> → <b>Plugins</b> → <b>Registry</b> → <b>IPhoneNormalizerPlugin.gwp</b>.</li> <li>Add a new parameter by clicking the + button to the right of the <b>Parameters</b> pane.</li> <li>For the <b>Name</b>, enter <code>extensionLength</code>.</li> <li>For the <b>Value</b>, enter a number representing the length of the phone number extension.</li> </ol>
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.
<b>Other - Database</b>	
PL-28631	Fixed an issue where <code>SystemToolsAPI.getUpdateStatsState</code> would return the wrong state.
<b>Other - Developer IDE</b>	
PL-27186	Studio now shows warnings for improper usages of internal types when the usage is in an <code>extends</code> or <code>implements</code> 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 <b>OSGi Bundle Facet</b> . To the right of Guidewire product directory text field, click the <b>Change</b> button and set the new disk path. The tool updates IDE library dependencies and <code>build.properties</code> .
<b>Other - Integration</b>	
PL-29002	This release adds a new servlet utility class <code>gw.servlet.ServletUtils</code> , 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.
<b>Other - Persistence</b>	
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.
<b>Plugins</b>	
PL-28621	Failing to start a startable plugin during server start now prevents the server from advancing to the next run level.
<b>Queries</b>	
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.
<b>Revisioning</b>	
PL-27210	Fixed an incorrect error that was generated by the "Verifies foreign key reference to a subtype is to correct subtype" consistency check.

ID	Description
<b>Templates</b>	
PL-27589	<p>The WS-I web service called TemplateToolsAPI changed to expand functionality to note and email templates, not just document templates:</p> <ul style="list-style-type: none"> <li>(a) The four methods validateAllTemplates, validateTemplate, validateTemplateInLocale, and listTemplate have new names validateAllDocumentTemplates, validateDocumentTemplate, validateTemplateDocumentInLocale, listDocumentTemplate. The old method names are deprecated.</li> <li>(b) New methods for note templates: validateAllNoteTemplates, validateAllNoteTemplatesInLocale, validateNoteTemplate, validateNoteTemplateInLocale, listNoteTemplate</li> <li>(c) New methods for email templates: validateAllEmailTemplates, validateAllEmailTemplatesInLocale, validateEmailTemplate, validateEmailTemplateInLocale, listEmailTemplate</li> </ul>
<b>Web - Configuration</b>	
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.
<b>Web - ListViews</b>	
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.
<b>Web - UI/Runtime</b>	
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 <b>Group By This Field</b> in a list view when it is in edit mode.
<b>Web Services - WSI (New)</b>	
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 run-level 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 section describes known issues with this release of Guidewire ClaimCenter:

- “ClaimCenter Known Issues” on page 179
- “Platform/Studio Known Issues” on page 180

**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 sections. The goal of this policy is to make upgrades as straightforward as possible.

### ClaimCenter Known Issues

#### Extending ActivityPattern results in an exception during upgrade (CLM-27857)

**Issue** – The `AddActivityPatternsFromImportDataVersionTrigger` upgrade trigger throws an exception if the `ActivityPattern` entity is extended.

Two scenarios can cause this exception:

- Case 1 – Adding a new column to `ActivityPattern` during the upgrade process.
- Case 2 – Adding a non-nullable column (`nullok = false`) to `ActivityPattern` during the upgrade process.

**Workaround** – The following steps can be taken, depending on the case.

- Case 1 – Add the column to the previous version of ClaimCenter before performing the major version upgrade. If this is not possible, add the column after the upgrade is complete.
- Case 2 – The column to be added must be nullable (`nullok = true`), and ClaimCenter must alter the column before the major version upgrade is attempted. You can change it back after the upgrade.

#### Too many compile warnings in ClaimCenter Studio (CLM-26978)

**Issue** – The number of warnings when compiling code in ClaimCenter Studio make it hard to see actual problems.

**Workaround** – Deprecation of RPC web services causes the largest number of warnings. If you are not using RPC web services, you can safely ignore warnings regarding `RpcWebService` or classes with `soap` in their package names.

A smaller number of warnings are related to the section inclusion limit for second pass verification of PCF files. The default limit is 1,000 for performance reasons, but a number of PCF files are more complicated than that. If you want to get rid of these warnings, you can change the setting manually in Guidewire Studio by navigating to `File → Settings → Guidewire Studio`. In the `Guidewire Studio` page, change the settings for `PCF Settings`. You can turn off `Limit Second Pass Verification` altogether or set `Section Inclusion Limit` to 3000, which is high enough for all the PCF files in the base configuration of ClaimCenter.

There is no workaround for the rest of these warnings.

#### FNOL Snapshot cannot support upgraded property claims (CLM-26997)

**Issue** – In the ClaimCenter 8.0.1 `FNOL Snapshot → Loss Details` PCFs, there is a reference to a property that did not exist in ClaimCenter 8.0.0. As a result, FNOL snapshots created for property claims in 8.0.0 cannot be viewed in 8.0.1.

**Workaround** – The workaround necessitates some changes in your PCF files. Edit `ClaimSnapshotGeneralPR800PanelSet.default.pcf`, `ClaimSnapshotGeneral800PanelSet.auto.pcf`, `ClaimSnapshotGeneral800PanelSet.g1.pcf`, and `ClaimSnapshotGeneral800PanelSet.Trav.pcf`, as follows:

- Replace FixedPropertyIncidentInstance.getProperty().Address.DisplayAddressLine1 with (FixedPropertyIncidentInstance.getProperty().Address.AddressLine1Kanji as String).HasContent ? FixedPropertyIncidentInstance.getProperty().Address.AddressLine1Kanji : FixedPropertyIncidentInstance.getProperty().Address.AddressLine1.
- Replace FixedPropertyIncidentInstance.getProperty().Address.DisplayCity with (FixedPropertyIncidentInstance.getProperty().Address.CityKanji as String).HasContent ? FixedPropertyIncidentInstance.getProperty().Address.CityKanji : FixedPropertyIncidentInstance.getProperty().Address.City.
- Replace prop.Property.Address.DisplayAddressLine1 with (prop.Property.Address.AddressLine1Kanji as String).HasContent ? prop.Property.Address.AddressLine1Kanji : prop.Property.Address.AddressLine1.
- Replace prop.Property.Address.DisplayCity with (prop.Property.Address.CityKanji as String).HasContent ? prop.Property.Address.CityKanji : prop.Property.Address.City.

### Date in Search by Contact results unclear (CLM-26978)

**Issue** – When you search for claim contacts using **Search by Contact**, the **Search Results** pane displays a date for search results that does not map to the Loss Date for the claim, as expected.

**Workaround** – None. Guidewire is aware of this issue.

### Issues with ISO DataPower Integration (CLM-27944)

**Issue** – ClaimCenter includes changes for supporting ISO’s DataPower platform. At the time of this release, there are a few unresolved issues that are being actively worked on by ISO. The original ISO integration remains intact as the default integration.

**Workaround** – None. Please contact Guidewire Support for the latest status of the ISO DataPower integration, if needed.

## Platform/Studio 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, ClaimCenter throws **ParseResultsException**. This is the intended behavior.

**Workaround** – Restart the workflow engine. To do so:

1. Log in to ClaimCenter 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 `gwXX regen-java-api`, an `index.html` file is not created in `ClaimCenter/java-api/doc`.

**Workaround** – The command now generates Javadoc JAR files in `ClaimCenter/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 gwcc 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 ClaimCenter/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 gwcc 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 ABCContact to Address for the primary address, and a one-to-one relationship from Address to ABCContact. 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** – 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 ClaimCenter 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 "gwxx 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 22

# Guidewire ClaimCenter 8.0.1 Release Notes

## Release 8.0.1

**Document Revision:** 23-June-2016

Copyright © 2001-2016 Guidewire Software, Inc.

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, Guidewire Spotlight, Gosu, Adapt and succeed, 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.

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.

## 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 ClaimCenter.

---

**IMPORTANT** If you skipped one or more upgrade releases to ClaimCenter, be sure to read the release notes for those releases to learn about changes and fixed issues.

---

### Version Number

This release of Guidewire ClaimCenter is 8.0.1.

- If you are integrating with Guidewire PolicyCenter, this version of ClaimCenter requires PolicyCenter 7.0.0 or later. The latest maintenance release is preferred.
- If you are integrating with Guidewire ContactManager, this version of ClaimCenter requires ContactManager 8.0.1 or later. The latest maintenance release is preferred.

## Installing This Release

Refer to the following:

- The *ClaimCenter Installation Guide* for general installation information
- Prior ClaimCenter 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. For information on new features and major changes, see the topic “New and Changed in ClaimCenter 8.0.1” in the ClaimCenter New and Changed Guide.

- Base PCF File Changes
- Base Rule Changes
- Changes in This Release Provided in Upgrade Diff Report

### 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.

#### **ClaimCenter release 8.0.0 to 8.0.1**

To view a report of the changes to the base PCF files, *[click here](#)*.

## Base Rule 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.

### ClaimCenter release 8.0.0 to 8.0.1

To view a report of the changes to the base rules, [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
<b>Application Server</b>	
PL-27203	JBoss 6 deployment now works properly.
<b>Archiving</b>	
PL-26930	Modified the <code>restoreSeveredTransactionOffsetOnsetOnsetLinks</code> method so it can be called from Gosu.
CLM-26553	In previous releases, the <code>PublicID</code> of a <code>Claim</code> was automatically synchronized with that of its associated <code>ClaimInfo</code> . When the two entities were linked, the <code>PublicID</code> would be copied from <code>ClaimInfo</code> to <code>Claim</code> . With this release, the values are no longer automatically synchronized. The <code>PublicIDs</code> for both entities are allowed to have their normal, generated values, which prevents possible collisions between <code>ClaimInfo</code> instances. Because of this change, we recommend that customer configurations be reviewed to ensure that they do not assume the <code>PublicIDs</code> will match between <code>Claim</code> - <code>ClaimInfo</code> pairs. To identify the <code>Claim</code> associated with a <code>ClaimInfo</code> , use <code>ClaimInfo.RootPublicID</code> instead.
<b>Archiving, Financials - Bulk Invoices</b>	
CLM-13501	All claims associated with bulk invoice items can now be archived, even if there are activities or history events for the bulk invoices. The approval history and history events for a <code>BulkInvoice</code> live on the claim of its first <code>BulkInvoiceItem</code> and will disappear when the claim is archived.
<b>Assignment, Email, Notes, Documents, Question Sets</b>	
CLM-25503	There was a problem with the Kanji Group field displaying on the <code>Assign</code> screen for assigning a claim when not in a Japanese locale. This field now displays only when your locale is Japan.
CLM-26204	Corrected an issue preventing a Note from being added that used the <code>ActionPlan</code> template.
CLM-26613	In the previous release, when a claim or exposure was closed, its associated weighted workload weight was set to zero. With this release, the workload weight of a closed claim or exposure retains the last calculated value. The user's adjusted weight is recalculated, excluding the closed claim or exposure. See "Workload Weight Recalculation" in the ClaimCenter Configuration Guide.

**Batch Processes**

PL-18406	<p>In releases prior to 8.0.0, you created custom batch processes by developing Gosu classes that implement the BatchProcess interface and subclass from the BatchProcessBase class. You configured custom batch processes in the scheduler-config.xml 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 process by developing Gosu classes that subclass from the WorkQueueBase class. You configure this newer type of custom batch process in the work-queue.xml file, with definitions for its worker processes that run on other servers.</p> <p>Note: The methods on WorkQueueBase 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>
----------	--

**Bundles and Transactions**

PL-22240	The touch method has been reimplemented. Please refer to the <i>Gosudoc</i> for detailed usage.
----------	---

**Catastrophe, Performance**

CLM-25255	Improved database performance of catastrophe heat maps via denormalization of Claim.LossLocation.SpatialPoint up to Claim.LossLocationSpatialDenorm.
-----------	--

**Claim File - Associations, Calendar, Claim Status, Evaluation, Exposures, History, Incidents, Loss Details, Matters, Negotiation, SIU, Summary**

CLM-16798	Visibility expressions have been removed for many state and jurisdiction fields in PCF files. These expressions previously controlled visibility of drop-down lists based on whether the default country had states, which was not always the correct condition for multi-country customers. Guidewire recommends that you configure your Jurisdiction.ttx and State.ttx typelists to have the values that you want to work with and retire any values that are unneeded.
-----------	---

**Claim - FNOL Wizard, New Claim Wizard, Claim Snapshot, FNOL mapper, Contacts, Roles**

CLM-26622	In previous releases, when you signed in as Super User ("su"), accessed the New Claim wizard, and added a contact to a claim from ContactManager that did not have the claimparty tag, a ConcurrentDataChangeException was thrown. In ClaimCenter 8.0.1, this has been fixed and the wizard can be completed as normal. The updates to the contacts will be accumulated through the wizard and sent to ContactManager after the claim is completed.
-----------	---

**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 where 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.

**Contacts, Roles**

CTC-2378	<p>Changed behavior of the Update button on the New Contact toolbar in ClaimCenter. This button checks for duplicate contacts only when both the following conditions are true:</p> <ul style="list-style-type: none"> <li>• The user has not already manually checked for duplicates.</li> <li>• The contact is expected to be stored in ContactManager, based on the ContactSystemApprovalUtil implementation.</li> </ul>
----------	---

CLM-24721	If ClaimCenter is integrated with ContactManager, when you create a new contact from the <b>Parties Involved</b> → <b>Contacts</b> screen, the edit screen now has a <b>Find Duplicates</b> button. Additionally, this screen automatically checks for duplicate contacts if you have not already clicked <b>Find Duplicates</b> . If you add a new contact and it is a duplicate of another contact, the ContactManager duplicate becomes the new contact. If the contact is only a potential duplicate of existing contacts in ContactManager, you can either select a contact from a list of potential duplicates or create the new contact.
CLM-26174	If ClaimCenter is integrated with ContactManager, the <b>New Claim</b> wizard and the <b>Check</b> wizard can now check for duplicate contacts. If you add a new vendor contact in ClaimCenter and it is a duplicate of another contact already in ContactManager, ClaimCenter displays a worksheet listing the duplicate contacts. If you select one of the duplicate contacts, the ContactManager duplicate becomes the new contact.
CLM-26542	In this release, there is now a commit-time enforcement in ClaimCenter that prevents multiple Address instances from being shared by two or more different contacts. A <code>Contact.PrimaryAddress</code> may not be referenced by any other <code>Contact.PrimaryAddress</code> or <code>Contact.SecondaryAddress</code> .  In previous releases, there was a database consistency check (DBCC) for such cases and now, a commit-time check has been added. The enforcement does not allow a shared reference to an address instance even when one of the referring <code>Contact</code> or <code>ContactAddress</code> instances is retired.  Customers who are upgrading to ClaimCenter 8.0.1 must ensure that their database does not contain any shared references to address instances. Refer to KB article #490, which has instructions for making necessary corrections. The corrections involve creating a duplicate Address for any shared instance and pointing each referring contact at the duplicate instead of the shared original.
CLM-26667	With this release, the <b>Primary phone</b> and <b>Mobile</b> fields are available for person vendor contacts.
<b>Contacts, Roles, Integration CM</b>	
CLM-16350	In previous releases of ClaimCenter, when a runtime exception occurred during the processing of a <code>ContactAutoSyncWorkItem</code> , it did not change to error status. This has now been fixed, and the status of the work item is appropriately changed to error.
CLM-26070	There were problems with creating related contacts for a contact in ClaimCenter, resulting either with no related contact being created in ContactManager, or, in some cases, actually deleting an existing contact in ContactManager. These problems have been fixed, and related contacts created in ClaimCenter now work correctly with ContactManager.
CLM-26098	There was a problem with contact search from the Address Book that prevented some related contacts from being returned. That problem has been corrected, and all related contacts are now visible in contact search results.
CLM-26789	In previous releases, when you created a contact in ClaimCenter that was to be synced in ContactManager, if validation for the creation failed in ContactManager, the contact was nevertheless created in ClaimCenter and remained unlinked. With this release, when you attempt to create a contact to be synced, if ContactManager validation fails, ClaimCenter displays an error message and does not complete local contact creation.
<b>Contacts, Roles, Integration CM - AutoSync, Contact Subtype Security, SPM, Driving Directions, Proximity Search</b>	
CLM-25971	Previously, the <code>config.xml</code> configuration parameter <code>InstantaneousContactAutoSync</code> was <code>false</code> by default, and ClaimCenter would synchronize contacts with ContactManager only when the <code>ContactAutoSync</code> batch process ran. This setting is now true by default in the base configuration. When ClaimCenter detects a change to a linked contact, its default behavior is to immediately synchronize the contact.
<b>Data Distribution</b>	
PL-25702	The Data Distribution page enables <b>Download comparison Zip file</b> and <b>Download Combined Zip file</b> options only if two executions are selected. In prior versions, the options were erroneously enabled even if there were not two executions selected.
<b>Database Configuration</b>	
PL-28656	Fixed a problem that prevented the <code>DBAuthenticationPlugin</code> from working.
<b>Database Support – General</b>	
PL-24417	The performance requirement to set the <code>action="delete"</code> attribute for <code>cc_message</code> in <code>database-config.xml</code> has been removed.
PL-28323	Fixed a bug which prevented scheduling of the Database Statistics process.

<b>Database Support – Oracle</b>	
PL-28327	Guidewire added an option to switch off the Oracle adaptive optimization feature for Guidewire applications. See “Configuring Oracle Adaptive Optimization for ClaimCenter” in the <i>ClaimCenter Installation Guide</i> .
<b>Database Support</b>	
PL-27438	Added support for Oracle Date interval partitioning.
<b>Database Upgrade</b>	
PL-27016	Guidewire has added support for using Oracle's parallel DDL execution feature during upgrade. The <code>createIndexInParallel</code> attribute of the <code>&lt;upgrade&gt;</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 <code>ID</code> column properly.
<b>DBChecks - Consistency/Integrity, Staging Tables, LoaderCallbacks, Constraints - Core</b>	
CLM-22975	<p>Database consistency checks, loader integrity checks, and commit-time enforcement have been added to verify the following:</p> <ul style="list-style-type: none"> <li>A policy that is not retired is always referred to by exactly one claim.</li> <li>The claim that refers to the policy is not retired.</li> </ul> <p>As in previous releases, policies and claims must be linked one-to-one. This condition is now enforced more strictly.</p>
CLM-24471	The class <code>CCQueryProcessorWithSummaryQuery</code> has been renamed <code>CCQueryBeanResultWithSummary</code> and moved from a <code>com.guidewire</code> package to the package <code>gw.cc.database</code> . Use this class to specify a second summary Query to populate the summary row of a query-backed LV.
CLM-25226	Corrected an issue where some loader integrity checks on <code>ClaimContactRole</code> did not report correct logical unit of work IDs (LUWIDs) when the integrity check failed.
CLM-26629	With this release, the database consistency check that verifies that <code>Policy.TotalVehicles</code> is at least as great as the number of <code>VehicleRU</code> entities linked to the policy was updated to ignore retired vehicles.
<b>Document Management</b>	
PL-27480	Document templates no longer have size constraints.
PL-27577	You can now call the method <code>DocumentsUtil.createNewDocument</code> without needing the current user to have the <code>doccreate</code> permission. Use this method if you implement the <code>IDocumentMetadataSource</code> plugin.
<b>Email</b>	
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.
<b>Entities/Metadata</b>	
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 where <code>MonetaryAmount</code> appeared as two separate fields under <code>Actual Amount</code> in the <i>Data Dictionary Data Entity View</i> .
PL-25809	Fixed an issue where typelist codes did not include documentation for the typecodes (name and description).
PL-27465	Added upgrade trigger to insert <code>xmlns</code> if missing on <code>.eti</code> files.

PL-27501	Added the ability to override/add keyfilters/typefilters via extensions.xml.
PL-27819	Added a new overlapTable attribute to the edgeForeignKey and localization entities, which specifies that the entity implements the OverlapTable delegate.
<b>Financials - Bulk Invoices</b>	
CLM-13732	When a bulk invoice is submitted in a ClaimCenter screen, the requesting user is now set to the current user, rather than the user who created the bulk invoice.
CLM-17832	Electronic Funds Transfer (EFT) fields were missing from the Bulk Invoice screens. These fields have now been added to those screens.
<b>Financials - Checks, Payments, Deductibles</b>	
CLM-16349	Individual checks in a recurring check set can now be transferred, subject to the same circumstances and conditions in which non-recurring checks were already allowed to be transferred. In particular, grouped checks are still not allowed to be transferred, whether in a recurring check set or not.
CLM-23239	A TransactionSet Approval rule was added to require approval when the exposure or claim of a check's payment is not at Ability To Pay validation level. Most of the time creating a check is prevented if Ability to Pay has not been reached. This rule catches checks created in the New Claim Wizard and through web services, and requires additional approval if not at Ability to Pay. The new rule is TAP04000 - Claim And Exposures Should Be At Ability To Pay.
<b>Financials - DBChecks - Consistency/Integrity, Staging Tables, LoaderCallbacks, Constraints</b>	
CLM-25826	Integrity checks now prevent recoveries from being imported that would cause total recoveries on a claim or exposure for a particular reserving currency to be negative.
<b>Financials - Multicurrency, Exchange Rates</b>	
CLM-22306	The payment and reserve rounding modes are now set to HALF_UP by default in config.xml. The configuration parameters are PaymentRoundingMode and ReserveRoundingMode. These configuration parameters are permanent and cannot be changed in production code. Therefore, if you are upgrading from a previous release, you must set these parameters to the settings in the previous release. If you are configuring an initial installation of ClaimCenter 8.0.1, you can leave these configuration parameter settings at HALF_UP. For more information, see "Choosing a Rounding Mode" in the Globalization Guide.
CLM-23453	Available reserves are now hidden in the Check wizard, the Recode screen, and the Transfer screen when a new reserve line is selected.
<b>Financials - ReserveLines, TAccounts, Financials Calculations, DBChecks - Consistency/Integrity, Staging Tables, LoaderCallbacks, Constraints</b>	
CLM-26055	Previously, Total Reserves and Total Eroding Payments per reserve line were not allowed to be negative. This restriction has been relaxed, and now Total Reserves and Total Eroding Payments must be non-negative for a given exposure and reserving currency combination, but not necessarily for a given reserve line.
<b>Globalization</b>	
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 where 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,
	<GWLanguge code="de_DE" name="German (Germany)" typecode="de_DE"> <ExtJsSettings> <LabelWidth size="190" /> </ExtJsSettings> </GWLanguge>

<b>Gosu</b>	
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.
<b>Integration</b>	
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 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. 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> .
<b>Integration CM</b>	
CLM-20466	There was a problem with the link status for a contact created in ClaimCenter that had an exact match to a contact in ContactManager. This problem has been fixed.
CLM-25261	There was a problem with ContactManager not deleting pending creates or pending updates for pending create entries when the associated contacts were deleted from ClaimCenter. The problem has been fixed. Now, if all claim contacts are deleted that caused a pending change or a pending create, ContactManager is notified and can remove the associated pending change or pending create.
<b>Integration CM - AutoSync, Contact Subtype Security, SPM, Driving Directions, Proximity Search, Core: Link, Related Contacts, Match, Retrieve, Search, Update</b>	
CLM-26068	Added a comment to the deprecated rule EFR08115 - Tags Changed indicating why it is deprecated in ClaimCenter 8.0.
<b>Integration ISO</b>	
CLM-25756	In this release, ISO integration uses two separate XSD files for requests and responses. All code paths creating the request to ISO use xsd.iso.req, and both synchronous and asynchronous responses from ISO are parsed using xsd.iso.resp.  The handleISOReply and sanityCheck methods in ISOReplyPlugin were changed to now accept a new ISOReplyPayload object instead of the XML and HttpRequest from ISO. The ISOReceive servlet places all parameters from the HTTP response into the ISOReplyPayload before it is passed to the ISOReplyPlugin.
<b>IntelliJ IDE – Compiler</b>	
PL-28346	Fixed a compilation error when compiling an entity or its extension if the entity had a subtype.
<b>IntelliJ IDE – Debugger</b>	
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.
<b>IntelliJ IDE – Display Key Editor</b>	
PL-27971	Improved typing performance in the Display Key editor.

PL-28677	Fixed a bug in Studio where converting a string into a display key caused exceptions. The behavior has changed slightly, so 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 the Step Name Localizations tab in the Workflow editor.
----------	--

**IntelliJ IDE – Entity Editor**

PL-26336	Column validation has been reenabled 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 where 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>&lt;tag&gt;</code> subelement of the <code>&lt;column&gt;</code> element, there is now a drop-down list showing available values.
PL-27454	Fixed an issue where 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 where pressing Ctrl+O 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 when pasting code into Studio where 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 then re-assign the variable to the result. The new operator <code>&amp;&amp;=</code> performs the logical AND operation to the previous value. The new operator <code>  =</code> performs the logical OR to the previous value. Both operators work with the primitive type <code>boolean</code> or the object type <code>Boolean</code> on either side of the operator. For example, suppose you have two <code>boolean</code> variables called <code>needsUpdate</code> and <code>flagTest</code> . The statement <code>needsUpdate   = flagTest</code> has the meaning of <code>needsUpdate = (needsUpdate OR flagTest)</code> . Do not confuse these new operators with the other operators <code>&amp;=</code> and <code> =</code> , 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 <code>for</code> statement after the <code>in</code> keyword. The code: <pre>for (x in 10) {...}</pre> is illegal and must be upgraded with an interval such as: <pre>for (x in 0.. 10) {...}</pre> using the provided inspection in Studio.

**IntelliJ IDE – Line of Business Editor**

PL-26590	The LOB tab has been removed from <code>.tti</code> 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 where Studio would throw an exception when a <code>categorylist</code> was added to a <code>LossType</code> typecode.

PL-26971	Fixed an issue in Studio where removing a typecode from its parent also incorrectly removed it from all of its other parents.
----------	---

**IntelliJ IDE – Other**

PL-27198	Fixed an issue where Run commands in the QuickJump box did not work when the server was started from Studio.
PL-27862	Fixed an issue where 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 – 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, ClaimCenter includes IntelliJ IDEA with OSGi Editor, an application separate from Guidewire Studio. For more information, refer to the <i>Integration Guide</i> .
----------	---

**IntelliJ IDE – Refresh**

PL-24108	Fixed an issue where the server threw an exception and did not handle newly created enhancements.
PL-28174	Fixed an issue where 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 where creating the first extension of a typelist caused an exception.
PL-26613	Fixed an issue in Studio where disabled typecodes looked enabled when selected.
PL-26535	Fixed an exception in the text editor of the typelist view.
PL-27174	Fixed an issue where the default setting for filtering metadata did not apply to the Typelist editor.
PL-27570	Fixed an issue where 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.
----------	--

**I18N, Language Pack, Localized Modules**

CLM-17936	With this release, the ClaimCenter calendar headings that previously used static date formats embedded in display keys have been updated to use the date formats for the appropriate locales.
CLM-24343	To support global phone numbers, the original single phone number String field was split into three separate fields: PhoneCountry, Phone (the national phone), and PhoneExtension. Policies retrieved from PolicyCenter have these three fields, and now the PolicyLocationSummary entity has been updated to support these three phone fields.

---

CLM-25537	<p>Drop-down lists showing typecodes from the <code>Jurisdiction.ttx</code> typelist are now filtered to show only jurisdictions related to a specific <code>JurisdictionType</code>, which varies by screen. You can configure these typelists and the categories relating them to control which jurisdictions appear in the drop-down lists on these screens.</p> <p>For example, when you are editing a contact who is a Person subtype in ClaimCenter, there is a <b>Driver's License</b> section on the edit screen. The <b>State</b> field on this screen uses <code>Jurisdiction.ttx</code> to populate the drop-down list for the field. If you want to change the contents of the drop-down list, edit the <code>Jurisdiction.ttx</code> file in Studio and add or remove values from the outgoing category <code>driving_lic</code>.</p>
CLM-26540	<p>The following question set fields are now localizable and enable question sets to be defined across multiple languages:</p> <ul style="list-style-type: none"> <li>• <code>Question.Text</code></li> <li>• <code>Question.DefaultAnswer</code></li> <li>• <code>QuestionChoice.Name</code></li> <li>• <code>QuestionChoice.Description</code></li> <li>• <code>QuestionSet.Name</code></li> <li>• <code>SubQuestion.Text</code></li> </ul>

**Other – Persistence**

PL-25820	Fixed an issue with whitespace not being trimmed by trimming unicode full-width whitespace.
----------	---

**Profiling**

PL-28172	Fixed an issue where the <code>Purge Profiler Data</code> 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>
----------	---

For example, the following Gosu sample code creates a row query, with `Address` as the primary entity. The result includes only the `City` column for `Address` instances that match the query criteria.

```
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))})
```

The `QuerySelectColumns` 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 `Address` instances.

```
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 addressQuery.withDistinct(true) // Always run aggregate queries
                                with Distinct set to true.
addressResult = addressQuery.select({QuerySelectColumns.dbFunction(
    DBFunction.Count(Paths.make(Address#City)))})
})
```

PL-27474	Fixed an issue where the <code>Count</code> property on a <code>gw.api.database.Query</code> result produced an incorrect SQL statement and count result when the <code>withDistinct(true)</code> setting was combined with a <code>Join</code> .
----------	---

<b>Search</b>	
PL-27359	A new parameter, <code>absolutePathToSortTmpDir</code> , has been added to the <code>batchload-config.xml</code> file. This parameter sets the directory into which the sort program used by the free-text batch load command writes its intermediate files. This directory may be the same as or different from the directory specified by the parameter <code>absolutePathToWorkDir</code> . This latter parameter still specifies the directory into which the free-text batch load command itself writes its intermediate files.
<b>Vendor Management - Services Directory, Service Requests</b>	
CLM-19651	There was a problem in the New Claim wizard's Services step, which was not showing the correct vehicle for coverage amounts for service types of rental, towing, and auto body repair. This problem has been fixed, and the correct vehicle is now shown with these service types.
CLM-19962	In the base configuration, the list views of services on the claim Summary screen and the claim Service Requests screen are now grouped by incident. You can change the order of services in the list views by ordering them on a different column.
CLM-25794	When you search for vendors providing services, such as in a Search Address Book pop-up in the Parties Involved screen or the Address Book, ClaimCenter now enables you to filter on specific services.
<b>Web - ListViews</b>	
PL-10908	Added the <code>groupedOnEnter</code> 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 <code>height</code> attribute has been added to the <code>ListViewPanel</code> 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 columns reordering when switching between filtering views in a list view.
<b>Web – UI/Runtime</b>	
PL-27884	The following new methods were added to the Javascript <code>gw.api.util</code> class: <code>getValue</code> , <code>setValue</code> , <code>getValues</code> , <code>setValues</code> . These allow you to get or set values of input elements. For example, you can use these in <code>TemplatePanel</code> 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.
PL-28749	Fixed a critical security vulnerability to persistent cross-site scripting attacks.
<b>Work Queues</b>	
PL-27301	Improved the server performance when selecting available work queue items.
PL-28696	Fixed an issue with <code>workItem</code> orphan detection during daylight saving time adjustment.
<b>XMLElement (and XSD types)</b>	
PL-27131	Gosu XML/XSD types now support use of circular <code>xs:include</code> references.

## Known Issues and Limitations

This section describes known issues with this release of Guidewire ClaimCenter:

- “ClaimCenter Known Issues” on page 197
- “Platform/Studio Known Issues” on page 198

**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 sections. The goal of this policy is to make upgrades as straightforward as possible.

## ClaimCenter Known Issues

### Too many compile warnings in ClaimCenter Studio (CLM-26978)

**Issue** – The number of warnings when compiling code in ClaimCenter Studio make it hard to see actual problems.

**Workaround** – Deprecation of RPC web services causes the largest number of warnings. If you are not using RPC web services, you can safely ignore warnings regarding `RpcWebService` or classes with `soap` in their package names.

A smaller number of warnings are related to the section inclusion limit for second pass verification of PCF files. The default limit is 1,000 for performance reasons, but a number of PCF files are more complicated than that. If you want to get rid of these warnings, you can change the setting manually in Guidewire Studio by navigating to **File → Settings → Guidewire Studio**. In the **Guidewire Studio** page, change the settings for **PCF Settings**. You can turn off **Limit Second Pass Verification** altogether or set **Section Inclusion Limit** to 3000, which is high enough for all the PCF files in the base configuration of ClaimCenter.

There is no workaround for the rest of these warnings.

### Ambiguous contact rejection at the end of workflow (CLM-27016)

**Issue** – When, in a ClaimCenter workflow, you create a contact in ClaimCenter that does not meet the minimum requirements in ContactManager, the contact is rejected at the end of the workflow rather than at the point of contact creation. Some examples are at the end of the New Claim wizard or service request creation workflow. This can potentially cause confusion regarding the cause of the error, especially in cases where there are multiple contacts.

**Workaround** – Return to the **Parties Involved** screen and correct the contact information to fulfill minimum requirements. However, if there are multiple contacts on the claim, this will require some investigation, because the errant contact is not identified.

### FNOL Snapshot cannot support upgraded property claims (CLM-26997)

**Issue** – In the ClaimCenter 8.0.1 **FNOL Snapshot → Loss Details** PCFs, there is a reference to a property that did not exist in ClaimCenter 8.0.0. As a result, FNOL snapshots created for property claims in 8.0.0 cannot be viewed in 8.0.1.

**Workaround** – The workaround necessitates some changes in your PCF files. Edit `ClaimSnapshotGeneralPR800PanelSet.default.pcf`, `ClaimSnapshotGeneral800PanelSet.auto.pcf`, `ClaimSnapshotGeneral800PanelSet.g1.pcf`, and `ClaimSnapshotGeneral800PanelSet.Trav.pcf`, as follows:

- Replace `FixedPropertyIncidentInstance.getProperty().Address.DisplayAddressLine1` with  
`(FixedPropertyIncidentInstance.getProperty().Address.AddressLine1Kanji as String).HasContent ? FixedPropertyIncidentInstance.getProperty().Address.AddressLine1Kanji : FixedPropertyIncidentInstance.getProperty().Address.AddressLine1`.
- Replace `FixedPropertyIncidentInstance.getProperty().Address.DisplayCity` with  
`(FixedPropertyIncidentInstance.getProperty().Address.CityKanji as String).HasContent ? FixedPropertyIncidentInstance.getProperty().Address.CityKanji : FixedPropertyIncidentInstance.getProperty().Address.City`.
- Replace `prop.Property.Address.DisplayAddressLine1` with  
`(prop.Property.Address.AddressLine1Kanji as String).HasContent ? prop.Property.Address.AddressLine1Kanji : prop.Property.Address.AddressLine1`.
- Replace `prop.Property.Address.DisplayCity` with `(prop.Property.Address.CityKanji as String).HasContent ? prop.Property.Address.CityKanji : prop.Property.Address.City`.

## Date in Search by Contact results unclear (CLM-26978)

**Issue** – When you search for claim contacts using **Search by Contact**, the **Search Results** pane displays a date for search results that does not map to the Loss Date for the claim, as expected.

**Workaround** – None. Guidewire is aware of this issue.

## Platform/Studio Known Issues

### Problem with regen-java-api command and JAR files (PL-16351)

**Issue** – If you run the `gwXX regen-java-api` command, ClaimCenter creates a `ClaimCenter/java-api/lib` directory and puts JAR files for Java development in this directory. However, ClaimCenter 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:

`ClaimCenter/admin/lib`

Copy them into the following directory:

`ClaimCenter/java-api/lib`

### Problem with running regen-java-api command with deprecated=true flag (PL-28992)

**Issue** – If you run the `gwcc 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 `ClaimCenter/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 `depends` to `dev-deploy`, as follows:

```
<target name="regen-java-api" depends="dev-deploy"
```

4. Save the file and 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 `@WsiPermission` annotations to change the permissions without changing the WSDL.

**Workaround** – Guidewire is aware of this issue.

### 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, ClaimCenter throws `ParseResultsException`. This is the intended behavior.

**Workaround** – Restart the workflow engine. To do so:

1. Log in to ClaimCenter 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 `gwXX regen-java-api`, an `index.html` file is not created in `ClaimCenter/java-api/doc`.

**Workaround** – The command now generates Javadoc JAR files in `ClaimCenter/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  
}
```

## Upgrade trigger for postOnChange on PCF widgets is not working in some cases (PL-27755)

**Issue** – In PCF files for some widgets, the ClaimCenter 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 ClaimCenter 8.0.0 upgrade tool:

1. Find all instances of widgets that did not have their `postOnChange` properties converted. For example, search the `ClaimCenter/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.

## 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.

#### [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.

#### [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.

---

## chapter 23

# Guidewire ClaimCenter 8.0.0 Release Notes

## Release 8.0.0

**Document Revision:** 23-June-2016

Copyright © 2001-2016 Guidewire Software, Inc.

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, Guidewire Spotlight, Gosu, Adapt and succeed, 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.

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.

## Overview

These release notes contain the following sections:

- “Release Information” on page 202
- “Installing This Release” on page 202
- “Support” on page 202

- “Issues and Major Changes” on page 202
- “Known Issues and Limitations” on page 203

## Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

---

**IMPORTANT** If you skipped one or more upgrade releases to ClaimCenter, be sure to read the release notes for those releases to learn about changes and fixed issues.

---

### Version Number

This release of Guidewire ClaimCenter is 8.0.0.

- If you are integrating with Guidewire PolicyCenter, this version of ClaimCenter requires PolicyCenter 7.0.0 or later. The latest maintenance release is preferred.
- If you are integrating with Guidewire ContactManager, this version of ClaimCenter requires ContactManager 8.0.0 or later. The latest maintenance release is preferred.

## Installing This Release

Refer to the following:

- The *ClaimCenter Installation Guide* for general installation information
- Prior ClaimCenter 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.

- “Base PCF File Changes” on page 202
- “Base Rule Changes” on page 203
- “Changes in This Release Provided in Upgrade Diff Report” on page 203

### 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.

#### **ClaimCenter release 7.0.5 to 8.0.0**

To view a report of the changes to the base PCF files, *click here*.

## Base Rule 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.

### ClaimCenter release 7.0.5 to 8.0.0

To view a report of the changes to the base rules, *[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 ClaimCenter:

- “ClaimCenter Known Issues” on page 203
- “Platform/Studio Known Issues” on page 204

### ClaimCenter Known Issues

#### Settings for `PaymentRoundingMode` and `ReserveRoundingMode` in the base configuration [\(CLM-22306\)](#)

**Issue** – The default setting for the `PaymentRoundingMode` configuration parameter in `config.xml` is `DOWN`, and for `ReserveRoundingMode` it is `UP`. These settings were a way to prevent the sum of the rounded claim-currency payments from exceeding the sum of the rounded claim-currency reserves. Now that ClaimCenter supports multicurrency reserves, it is less likely that these kinds of rounding errors will occur. Therefore, both these settings will be changing to `HALF_UP` in a future release.

**Workaround** –

---

**IMPORTANT** These configuration parameters are permanent and cannot be changed in production code. If you are upgrading from a previous release, you must keep the current settings for these parameters.

---

If you are configuring an initial installation of ClaimCenter 8.0.0, you can consider changing these configuration parameter settings to `HALF_UP`. Before making this change, ensure that you understand the possible effects by reading the topic “Choosing a Rounding Mode” in the Globalization Guide.

#### Too many compile warnings in ClaimCenter Studio [\(CLM-25855\)](#)

**Issue** – The number of warnings when compiling code in ClaimCenter Studio make it hard to see actual problems.

**Workaround** – Deprecation of RPC web services causes the largest number of warnings. If you are not using RPC web services, you can safely ignore warnings regarding `RpcWebService` or classes with `soap` in their package names.

A smaller number of warnings are related to the section inclusion limit for second pass verification of PCF files. The default limit is 1,000 for performance reasons, but a number of PCF files are more complicated than that. If you want to get rid of these warnings, you can change the setting manually in Guidewire Studio by navigating to **File → Settings → Guidewire Studio**. In the **Guidewire Studio** page, change the settings for **PCF Settings**. You can turn off **Limit Second Pass Verification** altogether or set **Section Inclusion Limit** to 3000, which is high enough for all the PCF files in the base configuration of ClaimCenter.

There is no workaround for the rest of these warnings.

## Platform/Studio Known 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 `gwXX regen-java-api` command, ClaimCenter creates a `ClaimCenter/java-api/lib` directory and puts JAR files for Java development in this directory. However, ClaimCenter 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:

`ClaimCenter/admin/lib`

Copy them into the following directory:

`ClaimCenter/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, ClaimCenter throws `ParseResultsException`. This is the intended behavior.

**Workaround** – Restart the workflow engine. To do so:

1. Log in to ClaimCenter 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 ClaimCenter *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 ClaimCenter 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 (`ClaimCenter8_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 ClaimCenter 8.0, you can reorder the columns of list views or change their width. These settings are then saved by ClaimCenter 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 ClaimCenter.

## Javadoc command does not generate index file in expected location (PL-27679)

**Issue** – When you run `gwcc regen-java-api`, an `index.html` file is not created in `ClaimCenter/java-api/doc`.

**Workaround** – The command now generates Javadoc JAR files in `ClaimCenter/java-api/doc`. To view the Javadoc, add the JAR files to the Studio project.

## XSD types do not compile in Gosu after a configuration upgrade (PL-27688)

**Issue** – XSD types in the ISO plugin are not compiling in Gosu after a configuration upgrade, especially after an upgrade from ClaimCenter 6.0.x.

**Workaround** – After upgrading to ClaimCenter 8.0, ensure that the following entries are in the `compatibility-xsd.xml` file:

```
<xsd namespace="xsd.acord"/>
<xsd namespace="xsd.xml-ns"/>
<xsd namespace="xsd.iso"/>
```

## Command to generate data dictionary fails if maxSPVInclusions option is specified (PL-27693)

**Issue** – The data dictionary is not generated when you run the `gwcc 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 ClaimCenter 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 ClaimCenter 8.0.0 upgrade tool:

1. Find all instances of widgets that did not have their `postOnChange` properties converted. For example, search the `ClaimCenter/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 24

# Guidewire ClaimCenter 7.0.6 Release Notes

## Release 7.0.6

**Document Revision:** 23-June-2016

Copyright © 2001-2016 Guidewire Software, Inc.

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, Guidewire Spotlight, Gosu, Adapt and succeed, 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:

- “Release Information” on page 210
- “Installing This Release” on page 210
- “Support” on page 210

- “Issues and Major Changes” on page 210
- “Improvements and General Issues” on page 212
- “Known Issues and Limitations” on page 213

## Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

### Version Number

This release of Guidewire ClaimCenter is 7.0.6.

## Installing This Release

Refer to the following:

- The *ClaimCenter Installation Guide* for general installation information
- Prior ClaimCenter 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 might affect your installation.

- “Base PCF File Changes” on page 210
- “Rules Changes” on page 210
- “Changes in This Release Provided in Upgrade Diff Report” on page 211

### Base PCF File Changes

All links below require that the `ReleaseNotes_files` directory be on your local disk in the same directory as this release notes file.

#### **ClaimCenter release 7.0.5 to 7.0.6**

- To view a report of the changes in the base PCF files in the `modules/cc` directory, [click here](#).
- To view a report of the changes in the base PCF files in the `modules/p1` directory, [click here](#).

### Rules Changes

#### **ClaimCenter release 7.0.5 to 7.0.6**

- There are no changes to the base rules in the `modules/cc` directory.

## 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

Following are the primary improvements and issues corrected in this release.

ID	Description
<b>Archiving</b>	
CLM-24489	A new entity, PurgedRootInfo, was added to record the public IDs of entities implementing the RootInfo delegate that are purged. In the case of ClaimCenter, the entity implementing RootInfo is ClaimInfo. When a Claim object is purged, its ClaimInfo is purged as well, and a PurgedRootInfo entity is now automatically created to record the ClaimInfo public ID. These entities are not automatically deleted, so a customer with a high purge volume may wish to create a batch process to delete old, unwanted PurgedRootInfo instances.
<b>Command-line Tools, Toolkit, Data Dictionary, External Entities, CC Batch Process</b>	
CLM-26600	Fixed an issue where the ClaimValidationWorkQueue could throw a ConcurrentModificationException when running in multithreaded mode.
<b>Contacts, Roles</b>	
CLM-26945	The AddressBookFingerprint field, previously in the Contact table, has been moved to its own table to improve performance during database upgrades.
<b>DBChecks - Consistency/Integrity, Staging Tables, LoaderCallbacks, Constraints - Core</b>	
CLM-25652	A new database consistency check (DBCC) was introduced to capture violations to Lines of Business (LOB) typelist configuration requirements.
<b>Desktop, Dashboard, Team, Quick Jump, Preferences, Statistics, Vacation</b>	
CLM-26277	The Open Claim Handler count under the Dashboard tab was counting users and groups who only had closed claims assigned to them. This has been fixed to only count handlers of open claims.
CLM-25729	The following improvements were made to proximity search to improve performance: <ul style="list-style-type: none"> <li>ClaimCenter now passes the value of the configuration parameter, ProximityRadiusSearchDefaultMaxResultCount, to ContactManager, which uses this value to restrict the maximum number of results to return for the proximity search.</li> <li>ClaimCenter now sends a unique identifier for a new proximity search when it makes the initial call to ContactManager. Any subsequent calls that ClaimCenter makes to ContactManager to retrieve additional results for the query use this unique identifier. ContactManager can use this unique identifier to return values from a cache rather than initiating another proximity search.</li> </ul>
CLM-25856	
<b>Financials - DBChecks - Consistency/Integrity, Staging Tables, LoaderCallbacks, Constraints</b>	
CLM-25825	ClaimCenter now has Database Consistency and Integrity Checks to catch Claims/Exposures for which Total Recoveries have become negative.
<b>Financials - Multicurrency, Exchange Rates, Integration PC</b>	
CLM-22851	PCClaimSearchIntegration API has been updated to send back Currency information (as a separate field) for all the amount fields on Claim that are exposed to PC. This change is intended to support the requirements of PC 8.0, which is multi-currency aware. Hence, any PC8 system integrating with a CC7 system need to use the 706 version of PCClaimSearchIntegrationAPI.wsdl.  Note that this change doesn't affect integration with PC7 systems in any way - they will just ignore the additional Currency fields in the API response.
<b>Tests/Internal - Builders, Sample data, Run Command, Demo</b>	

CLM-14793	Fixed code related to Financials Calculations and Assignable Queues, which would execute SQL queries that could not return any results due to an empty IN list. Now, the code does not execute a query at all in such cases.
CLM-14794	
CLM-14795	
CLM-14796	
CLM-14797	
CLM-14798	
CLM-14799	

## Known Issues and Limitations

This section describes known issues with this release of Guidewire ClaimCenter.

- “ClaimCenter Known Issues” on page 213
- “Studio/Platform Issues” on page 214

**Note:** For maintenance releases, Guidewire often defers fixing configuration issues if they require merging files during the upgrade. Workarounds to some of these issues are listed in the following sections. The goal of this policy is to make upgrades as straightforward as possible.

### ClaimCenter Known Issues

#### New Claim wizard not saving selected vehicle in some cases (CLM-13973)

**Issue** – The second step of the New Claim wizard is not saving your vehicle selection when you leave the screen and then come back. For example, you choose a vehicle and then click the **Policy Details** link, and when you click **Basic Info** on the left to return to this screen, the vehicle is no longer selected.

**Workaround** – Reselect the vehicle when you return to the wizard screen.

#### Upgrade from ClaimCenter 6 to ClaimCenter 7 does not work for SQL Server if MigrateToLargeIDsAndDatetime2 is set to true (CLM-17739)

**Issue** – During upgrade with SQL Server as the database, the `cc_tmppagglimitrpt` table fails to be upgraded if it is not empty and `MigrateToLargeIDsAndDatetime2` is `true`.

**Workaround** – If there is data in the table, `cc_tmppagglimitrpt`, truncate the table before beginning the upgrade by running the following SQL command:

```
DELETE FROM cc_tmppagglimitrpt
```

See “Enabling Migration to 64-bit IDs (SQL Server Only)” in the *ClaimCenter Upgrade Guide*.

#### ABContactPlugin now requires ContactManager 7.0.1 or later (CLM-17990)

**Issue** – The class `gw.plugin.addressbook.ab700.PluginRetrieveOperation`, which is used by `ABContactPlugin`, has a method that requires the `ABContactAPI` web service in ContactManager 7.0.1 or later. This method, `retrieveReplacementContact`, will cause errors if your ClaimCenter application continues to be integrated with ContactManager 7.0.0. Guidewire recommends that you upgrade to ContactManager 7.0.1 or later. If you cannot upgrade ContactManager, you must apply the workaround for this issue.

**Workaround** – Copy ClaimCenter/modules/cc/gsrc/gw/plugin/addressbook/ab700/PluginRetrieveOperation.gs to ClaimCenter/modules/configuration/gsrc/gw/plugin/addressbook/ab700/PluginRetrieveOperation.gs. Edit the copied file and remove the annotation @Readonly, the retrieveReplacementContact method, and all calls to the retrieveReplacementContact method. Then save the file.

### Printing a page containing a CheckBoxGroupInput widget can cause an error (CLM-21520)

**Issue** – Printing a page that has a CheckBoxGroupInput widget in it can cause an error if you print the current claim page. The error occurs if you navigate to Actions → Print Claim and select one of the Current claim page options to print the page.

**Workaround** – Navigate to Actions → Print Claim, select Custom, select Loss Details, and click Print to print the claim's Loss Details page only.

### Creating an unverified policy with a location risk in the new claim wizard causes errors preventing snapshot generation and printing (CLM-24729)

**Issue** – Viewing the policy on a claim snapshot can display a PCF compilation error similar to the following:

ClaimSnapshotPolicyScreen.700.pcf:61: Invalid 'def' expression: The type "snapshot.v700.LocationBasedRU" cannot be converted to "snapshot.v700.PolicyLocation".

**Workaround** – In ClaimSnapshotPolicyScreen.700.pcf, the ClaimSnapshotPolicyLocations700Screen card has a ListDetailPanel that declares a variable named LocationBasedRU of type snapshot.v700.LocationBasedRU. Change the type of this variable to snapshot.v700.PolicyLocation.

## Studio/Platform 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, Internet Explorer 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 an Internet Explorer 9 workaround that reduces these issues. In Internet Explorer 9, you can try to change the use of software or hardware rendering by toggling the Accelerated Graphics option on the Advanced tab of the Internet Options dialog.

### Studio Rules do not use logical capitalization for root object's name (PL-10740)

**Issue** – Rule set root objects are named with first letter lower-cased, which can look strange for variables like aBContact.

**Workaround** – Use the variable name as provided by Guidewire Studio.

### 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** – Entity localization table names have a shorter, 16-character limit than other entity tables. If the localization table name exceeds the 16-character limit, the error message indicates incorrectly that 18 characters are allowed.

**Workaround** – Ensure that the localization `tableName` property specified in the entity extension file is less than 16 characters.

## US-Locations.txt file with the US geodata from Great Data 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 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 `ClaimCenter/bin/gwpc regen-java-api` command, ClaimCenter creates a `ClaimCenter/java-api/lib` directory and puts JAR files for Java development in this directory. However, ClaimCenter 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:

`ClaimCenter/admin/lib`

Copy them into the following directory:

`ClaimCenter/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, ClaimCenter throws `ParseResultsException`. This behavior is intended.

**Workaround** – Restart the workflow engine. To do so:

1. Log in to ClaimCenter 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 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. ClaimCenter 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, ClaimCenter 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 in the resource pane on the left to find the resource in the resource hierarchy.

---

## chapter 25

# Guidewire ClaimCenter 7.0.5 Release Notes

## Release 7.0.5

**Document Revision:** 23-June-2016

Copyright © 2001-2016 Guidewire Software, Inc.

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, Guidewire Spotlight, Gosu, Adapt and succeed, 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:

- “Release Information” on page 218
- “Installing This Release” on page 218
- “Support” on page 218

- “Issues and Major Changes” on page 218
- “Improvements and General Issues” on page 220
- “Known Issues and Limitations” on page 222

## Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

### Version Number

This release of Guidewire ClaimCenter is 7.0.5.

## Installing This Release

Refer to the following:

- The *ClaimCenter Installation Guide* for general installation information
- Prior ClaimCenter 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 might affect your installation.

- “Base PCF File Changes” on page 218
- “Rules Changes” on page 218
- “Changes in This Release Provided in Upgrade Diff Report” on page 219

### Base PCF File Changes

All links below require that the `ReleaseNotes_files` directory be on your local disk in the same directory as this release notes file.

#### ClaimCenter release 7.0.4 to 7.0.5

- To view a report of the changes in the base PCF files in the `modules/cc` directory, [click here](#).
- To view a report of the changes in the base PCF files in the `modules/p1` directory, [click here](#).

### Rules Changes

#### ClaimCenter release 7.0.4 to 7.0.5

- There are no changes to the base rules in the `modules/cc` directory.

## 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

Following are the primary improvements and issues corrected in this release.

ID	Description
<b>Archiving</b>	
CLM-23810	In previous releases, when retrieving an archived claim, field validators prevented the restoration of archived entity graphs if an archived field contained data that violated field validation for its data type. With this release, ClaimCenter turns off field validation during the restoration of entity graphs from the archiving data store.
<b>Claim Metrics</b>	
CLM-23738	There was a problem with adding new claim metrics to existing claims. This problem has been fixed, and the process now works as described in the newly updated topic “Adding a New Claim Metric” in the <i>ClaimCenter Configuration Guide</i> .
<b>Command line tools</b>	
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.
<b>Contacts, Roles</b>	
CLM-23828	Previously, the ClaimCenter user interface allowed you to try to create a related contact for a user, such as a claims adjuster. This action is invalid and would cause an exception to be thrown. The ability to perform this action has been removed from the user interface.
<b>Database support</b>	
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 ClaimCenter database or for specific tables only. For more information, see “Configuring Oracle LOB Types” in the <i>ClaimCenter Installation Guide</i> .
<b>Database upgrade</b>	
CLM-24436 (PL-23686)	<p>Generating table statistics during upgrade is now 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 new option in the <code>config.xml</code> file. In the <code>&lt;database&gt;</code> block, there is an <code>&lt;upgrade&gt;</code> block that contains configuration information for the overall database upgrade. There is a new attribute in the <code>&lt;upgrade&gt;</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>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 <code>UpgradeInfo</code> 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 <code>UpgradeInfo</code> 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 <code>UpgradeInfo</code> page, you get a more detailed <code>UpgradeInfo</code> report.</p> <ul style="list-style-type: none"> <li>The <code>UpgradeInfo</code> report shows the value of the <code>updatestatistics</code> attribute at the time of upgrade.</li> <li>Additionally, the <code>UpgradeInfo</code> 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.</li> </ul> <p><b>Note:</b> The incremental mode might not capture all cases that would have been run during an upgrade with <code>updatestatistics=true</code>.</p>

PL-23504	<p>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.</p> <p>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, verify that 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.</p>
<b>Entities/Metadata</b>	
CLM-24336 (PL-24640)	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.
<b>Gosu</b>	
CLM-24237 (PL-24177)	Resolved an issue that caused the exists method to throw an exception when the where clause included a Guidewire boxed Boolean instead of the Java primitive type boolean.
CLM-24217 (PL-24095), CLM-24273 (PL-24478)	<p>Fixed 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>
<b>Financials – Checks, Payments, Deductibles</b>	
CLM-24208	Fixed a rare issue in which an IndexOutOfBoundsException exception could occur at the end of the Check wizard, inside the TransactionSet.submitForApproval method.
<b>Financials – Misc</b>	
CLM-24628	The Javadoc for the following data builders has been updated to more clearly indicate that the withStatus method must be used to properly setup T-accounts when creating test financial transactions: PaymentBuilder, ReserveBuilder, RecoveryBuilder, and RecoveryReserveBuilder. KB article 2121 provides more information and examples on the proper usage of these data builders.
	<b>Note:</b> Data builders are supported only for creating test data and must not be used in production code.
CLM-24630	<p>The following data builders related to financial transaction T-accounts have been deprecated: TAccountBuilder, TAccountTransactionBuilder, and TAccountLineItemBuilder.</p> <p>Instead of using these deprecated data builders, use the withStatus method on the data builders PaymentBuilder, ReserveBuilder, RecoveryBuilder, and RecoveryReserveBuilder to create test financial transactions.</p> <p><b>Note:</b> Data builders are supported only for creating test data and must not be used in production code.</p> <p>There is more information in the Javadoc for these data builders and in KB article 2121.</p>
<b>Financials – ReserveLines, TAccounts, Financials Calculations</b>	
CLM-21868	When using FinancialsCalculator, calling withReserveLine at the same time as withClaim, withExposure, or useClaimLevelReserves can return an unexpected amount. A warning is now logged saying that this amount might not be correct, and that such usage of FinancialsCalculator might be unsupported in the future.
<b>Integration</b>	
CLM-24335 (PL-23132)	There was a problem with starting a listening thread on JMS when the application server was the latest version of WebSphere. This problem has been fixed for WebSphere 7 and WebLogic 10.3.5 and later. Now you can use a listening thread with a startable plugin on these versions of WebSphere and WebLogic.
<b>Integration, ISO</b>	
CLM-23606	On November 30, 2012, ISO ClaimSearch implemented a change for XML format customers that applies only to ClaimSearch Medicare Section 111 participants. To support this change, ClaimCenter incorporated ISO stylesheet changes from ISO Stylesheet Version: 5.4.2 Release Date: 11-20-2012.

<b>Integration, Performance</b>	
CLM-23709	Previously, the default number of sender threads for ISO messages was one thread in the base configuration. Performance with this single thread was found to be very slow. Therefore, the default number of sender threads for ISO messages has been increased to five. You are encouraged to do your own testing to see what number of threads work best in your environment.
<b>Localization</b>	
CLM-24290 (PL-24462)	In previous releases, ClaimCenter 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, ClaimCenter handles old Imperial dates correctly.
<b>Management plugin</b>	
PL-16543	With this release, the counters <code>NumActiveDBConnections</code> and <code>NumIdleDBConnectionsmanagement</code> are integers instead of strings, so management tools that connect by using JMX can plot them now on graphs.
<b>Performance</b>	
CLM-24281	The ClaimCenter Message.eix file contains indexes to speed up the messaging subsystem when it searches for messages with a particular type of primary object, such as <code>Claim</code> or <code>Contact</code> . Performance testing found that the existing indexes could be improved. Therefore, improved indexes have been added to <code>Message.eix</code> , resulting in faster query plans and message processing.
<b>Persistence</b>	
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.
<b>Web - UI/Runtime</b>	
PL-23848	PCF input elements now render currency amounts so the scale and appscale parameters are honored. For example, if <code>appscale == 0</code> and the amount is the integer 1, the input element renders the amount as "1". If <code>appscale == 2</code> and the amount is the integer 1, the input element renders the amount as "1.00".
<b>Web services – WS-I</b>	
PL-23514	This release introduces a new method, <code>preExecute</code> , 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 <code>preExecute</code> method in your invocation handler, and 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 <code>preExecute</code> method.
PL-23721	To detect duplicate operations from external systems, add the new annotation <code>@WsICheckDuplicateExternalTransaction</code> 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 ClaimCenter.

- “ClaimCenter Known Issues” on page 223
- “Studio/Platform Issues” on page 224

**Note:** For maintenance releases, Guidewire often defers fixing configuration issues if they require merging files during the upgrade. Workarounds to some of these issues are listed in the following sections. The goal of this policy is to make upgrades as straightforward as possible.

## ClaimCenter Known Issues

### New Claim wizard not saving selected vehicle in some cases (CLM-13973)

**Issue** – The second step of the New Claim wizard is not saving your vehicle selection when you leave the screen and then come back. For example, you choose a vehicle and then click the **Policy Details** link, and when you click **Basic Info** on the left to return to this screen, the vehicle is no longer selected.

**Workaround** – Reselect the vehicle when you return to the wizard screen.

### Upgrade from ClaimCenter 6 to ClaimCenter 7 does not work for SQL Server if MigrateToLargeIDsAndDatetime2 is set to true (CLM-17739)

**Issue** – During upgrade with SQL Server as the database, the `cc_tmppagglimitrpt` table fails to be upgraded if it is not empty and `MigrateToLargeIDsAndDatetime2` is `true`.

**Workaround** – If there is data in the table `cc_tmppagglimitrpt`, truncate the table before beginning the upgrade by running the following SQL command:

```
DELETE FROM cc_tmppagglimitrpt
```

### ABContactPlugin now requires ContactManager 7.0.1 or later (CLM-17990)

**Issue** – The class `gw.plugin.addressbook.ab700.PluginRetrieveOperation`, which is used by `ABContactPlugin`, has a method that requires the `ABContactAPI` web service in ContactManager 7.0.1 or later. This method, `retrieveReplacementContact`, will cause errors if your ClaimCenter application continues to be integrated with ContactManager 7.0.0. Guidewire recommends that you upgrade to ContactManager 7.0.1 or later. If you cannot upgrade ContactManager, you must apply the workaround for this issue.

**Workaround** – Copy `ClaimCenter/modules/cc/gsrc/gw/plugin/addressbook/ab700/PluginRetrieveOperation.gs` to `ClaimCenter/modules/configuration/gsrc/gw/plugin/addressbook/ab700/PluginRetrieveOperation.gs`. Edit the copied file and remove the annotation `@ReadOnly`, the `retrieveReplacementContact` method, and all calls to the `retrieveReplacementContact` method. Then save the file.

### Printing a page containing a CheckBoxGroupInput widget can cause an error (CLM-21520)

**Issue** – Printing a page that has a `CheckBoxGroupInput` widget in it can cause an error if you print the current claim page. The error occurs if you navigate to **Actions** → **Print Claim** and select one of the **Current claim page** options to print the page.

**Workaround** – Navigate to **Actions** → **Print Claim**, select **Custom**, select **Loss Details**, and click **Print** to print the claim's Loss Details page only.

### Creating an unverified policy with a location risk in the new claim wizard causes errors preventing snapshot generation and printing (CLM-24729)

**Issue** – Viewing the policy on a claim snapshot can display a PCF compilation error similar to the following:

```
ClaimSnapshotPolicyScreen.700.pcf:61: Invalid 'def' expression: The type "snapshot.v700.LocationBasedRU" cannot be converted to "snapshot.v700.PolicyLocation".
```

**Workaround** – In `ClaimSnapshotPolicyScreen.700.pcf`, the `ClaimSnapshotPolicyLocations700Screen` card has a `ListDetailPanel` that declares a variable named `LocationBasedRU` of type `snapshot.v700.LocationBasedRU`. Change the type of this variable to `snapshot.v700.PolicyLocation`.

## Studio/Platform 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, Internet Explorer 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 an Internet Explorer 9 workaround that reduces these issues. In Internet Explorer 9, you can try to change the use of software or hardware rendering by toggling the **Accelerated Graphics** option on the **Advanced** tab of the **Internet Options** dialog.

### Studio Rules do not use logical capitalization for root object's name (PL-10740)

**Issue** – Rule set root objects are named with first letter lower-cased, which can look strange for variables like `aBContact`.

**Workaround** – Use the variable name as provided by Guidewire Studio.

### 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** – Entity localization table names have a shorter, 16-character limit than other entity tables. If the localization table name exceeds the 16-character limit, the error message indicates incorrectly that 18 characters are allowed.

**Workaround** – Ensure that the localization `tableName` property specified in the entity extension file is less than 16 characters.

### US-Locations.txt file with the US geodata from Great Data 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 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 `ClaimCenter/bin/gwpc regen-java-api` command, ClaimCenter creates a `ClaimCenter/java-api/lib` directory and puts JAR files for Java development in this directory. However, ClaimCenter 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:

`ClaimCenter/admin/lib`

Copy them into the following directory:

`ClaimCenter/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, ClaimCenter throws `ParseResultsException`. This behavior is intended.

**Workaround** – Restart the workflow engine. To do so:

1. Log in to ClaimCenter 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. ClaimCenter 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, ClaimCenter 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 in the resource pane on the left to find the resource in the resource hierarchy.

# Guidewire ClaimCenter 7.0.4 Release Notes

## Release 7.0.4

### Release Notes Update: 13-February-2013

---

**IMPORTANT** This release note document adds a release note for issue CLM-13821 to the table at “Improvements and General Issues” on page 230. 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:** 23-June-2016

Copyright © 2001-2016 Guidewire Software, Inc.

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, Guidewire Spotlight, Gosu, Adapt and succeed, 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:

- “Release Information” on page 228
- “Installing This Release” on page 228
- “Support” on page 228
- “Issues and Major Changes” on page 228
- “Improvements and General Issues” on page 230
- “Known Issues and Limitations” on page 233

## Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

### Version Number

This release of Guidewire ClaimCenter is 7.0.4.

## Installing This Release

Refer to the following:

- The *ClaimCenter Installation Guide* for general installation information
- Prior ClaimCenter 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 might affect your installation.

- “Base PCF File Changes” on page 228
- “Rules Changes” on page 229
- “Changes in This Release Provided in Upgrade Diff Report” on page 229

### Base PCF File Changes

All links below require the `ReleaseNotes_files` directory on your local disk, in the same directory as this release notes file.

#### ClaimCenter release 7.0.3 to 7.0.4

- To view a report of the changes in the base PCF files in the `modules/cc` directory, *click here*.
- To view a report of the changes in the base PCF files in the `modules/p1` directory, *click here*.

## Rules Changes

### ClaimCenter release 7.0.3 to 7.0.4

- To view a report of the changes in the base rules in the `modules/cc` 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

Following are the primary improvements and issues corrected in this release.

ID	Description
<b>Archiving, Command-line Tools, Toolkit, Data Dictionary, External Entities, CC Batch Process</b>	
CLM-17203	<p>Fixed a Null Pointer Exception that resulted when processing a <code>ClaimException</code> work item that pointed to an archived claim. Previously, if a claim was archived when it still had a <code>ClaimException</code> work item to process, a Null Pointer Exception would result when the <code>ClaimException</code> work item was eventually processed.</p> <p>Now a warning message is logged instead.</p>
<b>Authentication</b>	
PL-21737	The Gosu implementation of the <code>AuthenticationSourceCreator</code> plugin now supports JBoss.
<b>Build Infrastructure</b>	
PL-15509	<p>In previous versions, Guidewire supported running build scripts only through the <code>gwcc</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 ClaimCenter/modules/ant/build.xml buildScript</pre> <p>Substitute one of the following values for <code>buildScript</code>:</p> <ul style="list-style-type: none"> <li>• <code>build-jboss-war</code> – Builds a generic WAR file for use with JBoss.</li> <li>• <code>build-tomcat-war</code> – Builds a generic WAR file for use with Tomcat.</li> <li>• <code>build-weblogic-ear</code> – Builds an EAR file for use with WebLogic.</li> <li>• <code>build-websphere-ear</code> – Builds an EAR file for use with WebSphere.</li> </ul> <p>Although you can run many build scripts by using the <code>gwcc</code> command on Windows, if you invoke Ant on Unix, Guidewire supports only the build scripts in the previous list.</p>
<b>Claim – FNOL Wizard, New Claim Wizard, Claim Snapshot, FNOL mapper</b>	
CLM-17694	<p>There was an issue with switching from the Auto First and Final claim wizard to the full New Claim wizard. Auto First and Final automatically creates an exposure with the selected vehicle incident, but the full New Claim wizard does not. If you started from Auto First and Final, switched to the full wizard, and then changed or removed incidents in Step 2, referential integrity errors could occur due to references from the no-longer-used exposures. This issue has been fixed by removing the automatically created exposures when switching to the full wizard.</p>
<b>Claim File – Associations, Calendar, Claim Status, Evaluation, Exposures, History, Incidents, Loss Details, Matters, Negotiation, SIU, Summary</b>	
CLM-21260	Changed the <code>ClaimInAssociation</code> entity to allow setting its <code>PrimaryClaim</code> field from Gosu code.
<b>Claim Metrics</b>	
CLM-17625	Corrected an issue where the <code>TimeToFirstPayment</code> metrics for claims and exposures did not consider manual checks.
<b>Cognos integration</b>	
PL-22539	With this release, the Cognos plugin no longer removes domain names from the URL parameter to the <code>setCognosCookiesUrl</code> plugin method.
<b>Command-line Tools</b>	
PL-20110	The command line utility <code>gwcc regen-soap-api</code> no longer fails on Japanese Windows.
<b>Command-line Tools, Toolkit, Data Dictionary, External Entities, CC Batch Process</b>	
CLM-21646	<p>Fixed an issue in which a custom authority limit profile might have appeared as a shared profile in multi-locale environments. To fix this problem, a boolean database column named <code>Custom</code> has been added to <code>AuthorityLimitProfile</code> to indicate whether a profile is a custom profile. Additionally, there is an upgrade trigger to populate this column. If the profile is used by more than one user, the trigger sets the column value to <code>false</code>, regardless of the name of the profile. Otherwise, the trigger sets the column value to <code>true</code> if the name of the profile is <code>Custom</code>. It also sets the value to <code>true</code> if the name of the profile matches the display key value of <code>Java.AuthorityLimitProfiles.CustomProfile.Name</code> in the default locale.</p>
<b>Contacts, Roles</b>	

CLM-21378	<p>There was an issue in which an array property on a Contact that was specified as <code>IgnoreProperty</code> would cause an error if the user added an element to the array in the Address Book tab of ClaimCenter. This issue has been fixed by not sending the new elements to Contact Manager, which was the behavior in releases prior to ClaimCenter 7.</p> <p>As a reminder, if you add an existing property to the <code>IgnoreProperty</code> list in ClaimCenter 7, you must also remove the <code>to XML</code> setting for the property from the Contact mapping file. Doing so prevents the property from being mapped to ContactManager, which matches the behavior of ClaimCenter in releases prior to version 7.</p>
CLM-22068	Contact information now correctly reflects what has been selected from the Address Book when creating a document from a document template.
CLM-22746	Fixed an issue in which an error in creating or updating a contact in the Address Book could leave newly created entities in ClaimCenter that had incorrect temporary public IDs.
CLM-22932	Fixed an issue in which <code>IgnoreProperty</code> entries in <code>contact-sync-config.xml</code> subtypes were not being honored for entities other than Contact.

#### Database

PL-21554	<p>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 <code>IdentifyQueriesViaComments</code> parameter in <code>config.xml</code> to provide comments with contextual information in certain SQL Select statements sent to the relational database.</p> <p>The SQL comments are in the format:</p> <p style="padding-left: 40px;"><i>ApplicationName:ActionName</i></p> <p><i>ApplicationName</i> is ClaimCenter.</p> <p><i>ActionName</i> is the name of the PCF file that submitted the SQL Select statement.</p>
----------	---

#### Database Upgrade

PL-22007	In earlier versions, ClaimCenter ran the trigger <code>AfterUpgradeVersionTrigger</code> on fresh databases, which could cause problems. With this release, ClaimCenter runs this trigger only on existing databases, not on fresh databases.
PL-22478	In earlier versions, on SQL Server queries failed if they contained a large number of CASE conditions. Now, ClaimCenter breaks up queries with more than 125 CASE conditions into nested CASE clauses or into separate queries.
PL-22686	With this release, the validation SQL run by the upgrade version checks is now included 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 of the <code>InstrumentedBatchJobID</code> column on the <code>cc_instrumentedworkertask</code> table.
PL-23044	With this release, the database upgrader 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.

#### Document Management

PL-13904	A new class, <code>DocTreeNode</code> , simplifies the creation of a tree based on a customer-defined path used to organize documents.
----------	--

PL-22562	<p>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.</p> <p>For example, to add conditional processing to the production of Microsoft Word documents:</p> <ol style="list-style-type: none"> <li><b>1. Declare the variable in the XSD for document template descriptors</b> – In Guidewire Studio, open the file document-template.xsd. Locate this XSD file by pressing CTRL+N and entering document-template. Add the variable as an XSD attribute, as the following sample code shows.</li> </ol> <pre>&lt;xsd:attribute name="mail-merge-type" type="xsd:string" use="optional"/&gt;</pre> <p>Then, save the file.</p> <ol style="list-style-type: none"> <li><b>2. Set the variable to a value in the document template descriptor</b> – In Guidewire Studio, navigate to resources → Other Resources → doctemplates and open the file SampleLetter.gosu.rtf.descriptor. Add the variable as an attribute on the &lt;DocumentTemplateDescriptor&gt; element, as the following sample code shows.</li> </ol> <pre>mail-merge-type="catalog"</pre> <p>Then, save the file.</p> <ol style="list-style-type: none"> <li><b>3. Use the variable in the Mail Merge and Close merge script</b> – In Guidewire 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.</li> </ol> <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 release includes an updated BFO library that corrects the problem.
<b>Entities/Metadata</b>	
PL-22866	Calling an API method that attempts to modify a read-only bundle now shows an immediate error.
PL-22597	In earlier versions of ClaimCenter 7, if you defined an entity type that implemented a Gosu interface, problems occurred if the interface had a method that returned 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.
<b>Financials – Aggregate Limits, Policy Period</b>	
CLM-22169	Fixed an issue in which changing a payment's exposure did not correctly update applicable aggregate limits.
<b>Financials – Checks, Payments, Deductibles</b>	
CLM-13821	Added a new method called resetCheckAndPaymentsToDraft on the CheckSet entity. If an edit is made to the check that requires reapproval, such as an edit to an extension field, you can call this method and the check will be resubmitted for approval when the wizard is finished.
<p><b>Note:</b> Editing the payee or amount of a check already triggers reapproval.</p> <p>You can call this new method in the beforeCommit attribute of the EditCheckWizard.pcf file. You must enter this method call before the call to the wizard.performBeforeFinish method. The resetCheckAndPaymentsToDraft method works inside the Check wizard only. It is not possible to edit the amount or other important fields of a saved check in rules outside the Edit Check wizard.</p>	
<p>As part of this change, ClaimCenter is now more careful about not letting transactions in draft status be saved to the database. If you receive the exception message Draft Transactions cannot be saved, you might need to fix your creation of test data or look for reasons why your configuration might be attempting to save transactions as draft.</p>	
<p>See Knowledge Base article 2121 for more information. This article is on the Guidewire Customer Portal at the following link:</p> <p><a href="https://guidewire.custhelp.com/app/answers/detail/a_id/2121">https://guidewire.custhelp.com/app/answers/detail/a_id/2121</a></p>	
CLM-15285	Updated the Check Creator to validate and provide a descriptive message when the primary check payee is not specified (is null).

CLM-16951	The starting date, time, and day in the description for a recurrence check is now localized.
CLM-17355	There was an issue with not being able to delete a check if attempting to delete the check caused validation warnings. This issue has been fixed. You can now delete a check from either the <b>Check Details</b> page or the <b>Payment Details</b> page even if there is a validation warning shown when performing the delete action.  <b>Note:</b> To take this fix, merge in changes to the files <code>ClaimFinancialsChecksDetail.pcf</code> and <code>DeleteSinglePayment.pcf</code> .
<b>Financials – Bulk Invoice</b>	
CLM-22977	Disallowing setting <code>BulkInvoice.PaymentMethod</code> to <code>manual</code> and setting <code>Check.PaymentMethod</code> to <code>manual</code> on a placeholder check for a <code>BulkInvoiceItem</code> object. These settings are not tested or supported, and have led to stuck bulk invoices and to integration problems. Be sure to merge in the change to <code>BulkInvoiceDetailDV.pcf</code> that removes the <code>BulkInvoice.PaymentMethod</code> field from the PCF page, so that users do not accidentally choose the illegal <code>manual</code> option.
<b>Financials – Miscellaneous</b>	
CLM-23162	A new implementation of the <code>IExchangeRateSetPlugin</code> plugin interface, <code>SampleExchangeRateSetPlugin2.gs</code> , is now the default implementation. The previous plugin implementation referenced a web service that is no longer in service. The new plugin implementation does not reference a web service. Instead, it simulates rate changes to enable demonstrations showing how exchange rates work in ClaimCenter.  See Knowledge Base article 2068 for information on the new plugin implementation.
<b>Gosu</b>	
PL-22469	When an entity contains an <code>implementsInterface</code> element, that entity type is supposed to implement the interface specified in the <code>implementsInterface</code> 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.
<b>I18N, Language Pack, Localized Modules</b>	
CLM-20543	Previously, database collations were missing the Italian locale. Database collations are now included for Italian.
<b>Web - Other</b>	
PL-18559	Added Alt text, alternative tooltip text, for the following: <ul style="list-style-type: none"><li>• <code>BooleanRadioInput</code></li><li>• <code>PickerLink</code></li><li>• <code>DocumentationCell</code></li></ul>

## Known Issues and Limitations

This section describes known issues with this release of Guidewire ClaimCenter.

- “ClaimCenter Known Issues” on page 234
- “Studio/Platform Issues” on page 234

**Note:** For maintenance releases, Guidewire often defers fixing configuration issues if they require merging files during the upgrade. Workarounds to some of these issues are listed in the following sections. The goal of this policy is to make upgrades as straightforward as possible.

## ClaimCenter Known Issues

### New Claim wizard not saving selected vehicle in some cases (CLM-13973)

**Issue** – The second step of the New Claim wizard is not saving your vehicle selection when you leave the screen and then come back. For example, you choose a vehicle and then click the **Policy Details** link, and when you click **Basic Info** on the left to return to this screen, the vehicle is no longer selected.

**Workaround** – Reselect the vehicle when you return to the wizard screen.

### Upgrade from ClaimCenter 6 to ClaimCenter 7 does not work for SQL Server if MigrateToLargeIDsAndDatetime2 is set to true (CLM-17739)

**Issue** – During upgrade with SQL Server as the database, the `cc_tmppagglimitrpt` table fails to be upgraded if it is not empty and `MigrateToLargeIDsAndDatetime2` is `true`.

**Workaround** – If there is data in the table `cc_tmppagglimitrpt`, truncate the table before beginning the upgrade by running the following SQL command:

```
DELETE FROM cc_tmppagglimitrpt
```

### ABContactPlugin now requires ContactManager 7.0.1 or later (CLM-17990)

**Issue** – The class `gw.plugin.addressbook.ab700.PluginRetrieveOperation`, which is used by `ABContactPlugin`, has a method that requires the `ABContactAPI` web service in ContactManager 7.0.1 or later. This method, `retrieveReplacementContact`, will cause errors if your ClaimCenter application continues to be integrated with ContactManager 7.0.0. Guidewire recommends that you upgrade to ContactManager 7.0.1 or later. If you cannot upgrade ContactManager, you must apply the workaround for this issue.

**Workaround** – Copy `ClaimCenter/modules/cc/gsrc/gw/plugin/addressbook/ab700/PluginRetrieveOperation.gs` to `ClaimCenter/modules/configuration/gsrc/gw/plugin/addressbook/ab700/PluginRetrieveOperation.gs`. Edit the copied file and remove the annotation `@ReadOnly`, the `retrieveReplacementContact` method, and all calls to the `retrieveReplacementContact` method. Then save the file.

### Printing a page containing a CheckBoxGroupInput widget can cause an error (CLM-21520)

**Issue** – Printing a page that has a `CheckBoxGroupInput` widget in it can cause an error if you print the current claim page. The error occurs if you navigate to **Actions** → **Print Claim** and select one of the **Current claim page** options to print the page.

**Workaround** – Navigate to **Actions** → **Print Claim**, select **Custom**, select **Loss Details**, and click **Print** to print the claim's Loss Details page only.

## Studio/Platform 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, Internet Explorer 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 an Internet Explorer 9 workaround that reduces these issues. In Internet Explorer 9, you can try to change the use of software or hardware rendering by toggling the **Accelerated Graphics** option on the **Advanced** tab of the **Internet Options** dialog.

## Studio Rules do not use logical capitalization for root object's name (PL-10740)

**Issue** – Rule set root objects are named with first letter lower-cased, which can look strange for variables like `aBContact`.

**Workaround** – Use the variable name as provided by Guidewire Studio.

## 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** – Entity localization table names have a shorter, 16-character limit than other entity tables. If the localization table name exceeds the 16-character limit, the error message indicates incorrectly that 18 characters are allowed.

**Workaround** – Ensure that the localization `tableName` property specified in the entity extension file is less than 16 characters.

## US-Locations.txt file with the US geodata from Great Data 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 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 `ClaimCenter/bin/gwpc regen-java-api` command, ClaimCenter creates a `ClaimCenter/java-api/lib` directory and puts JAR files for Java development in this directory. However, ClaimCenter 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:

`ClaimCenter/admin/lib`

Copy them into the following directory:

`ClaimCenter/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, ClaimCenter throws `ParseResultsException`. This behavior is intended.

**Workaround** – Restart the workflow engine. To do so:

1. Log in to ClaimCenter using an administrative account.
2. Access Internal Tools → Reload.
3. Click Reload Workflow Engine.

---

## chapter 27

# Guidewire ClaimCenter 7.0.3 Release Notes

## Release 7.0.3

**Document Revision:** 23-June-2016

Copyright © 2001-2016 Guidewire Software, Inc.

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, Guidewire Spotlight, Gosu, Adapt and succeed, 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:

- “Release Information” on page 238
- “Installing This Release” on page 238
- “Support” on page 238

- “Issues and Major Changes” on page 238
- “Improvements and General Issues” on page 240
- “Known Issues and Limitations” on page 243

## Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

### Version Number

This release of Guidewire ClaimCenter is 7.0.3.

## Installing This Release

Refer to the following:

- The *ClaimCenter Installation Guide* for general installation information
- Prior ClaimCenter 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 might affect your installation.

- “Base PCF File Changes” on page 238
- “Rules Changes” on page 238
- “Changes in This Release Provided in Upgrade Diff Report” on page 239

### Base PCF File Changes

All links below require the `ReleaseNotes_files` directory on your local disk.

#### **ClaimCenter release 7.0.2 to 7.0.3**

- To view a report of the changes in the base PCF files in the `modules/cc` directory, [click here](#).
- To view a report of the changes in the base PCF files in the `modules/p1` directory, [click here](#).

### Rules Changes

#### **ClaimCenter release 7.0.2 to 7.0.3**

- To view a report of the changes in the base rules in the `modules/cc` 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

Following are the primary improvements and issues corrected in this release.

ID	Description
<b>Activities, Activity Patterns, Workplan</b>	
CLM-20883	Fixed an issue in which viewing activities assigned to a queue would throw an exception.
<b>Archiving</b>	
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. A more informative error message is now displayed.
PL-21380	The regen-xsd command generated an XSD file with errors. The command has been updated to produce a valid XSD file.
<b>Assignment, Email, Notes, Documents, Question Sets</b>	
CLM-15535	Corrected an issue in which if ActiveX was disabled, a new document failed to show up in the ActivityDocumentsLV of the ActivityDetailWorksheet.
<b>Batch Processes</b>	
PL-20755	Fixed an issue in which worker threads could unexpectedly quit if any database or JDBC error occurred during a time span longer than the maximum wait period.
<b>Command Line Tools</b>	
PL-19768	Fixed an issue that occurred while attempting to execute the verify-types command on a 64-bit JVM, which caused an OutOfMemoryError.
<b>Consistency Checker</b>	
PL-16205	If you request a run of the database consistency checks using the <database> element attribute checker="true" (in config.xml), ClaimCenter now prints a warning message in the log warning that this feature is deprecated. The message also instructs you to use the Info Page or command line to run the consistency checks as a batch job.
<b>Database</b>	
PL-8729	Guidewire removed the configuration parameter TableEstimatePercent from the config.xml file. Instead, use the samplingpercentage attribute of the database statistics element in config.xml, 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-17742	On Oracle, changed database upgrade logic by providing additional optimizer hints to help improve upgrade performance.
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-21448	On SQL Server, ClaimCenter now provides additional information on database connections, which can then be captured and analyzed in the Server DMV Snapshot screen.

**Database Support**

- PL-19689 Guidewire has changed the mechanism for creating the database performance reports for Oracle and SQL Server. You no longer have to click a button and wait 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 cc\_dbperfreport. This change requires a database upgrade to create this new table.

**Database Upgrade**

- PL-16977 Previously after a database upgrade, ClaimCenter logged differences between the metadata configuration and the upgraded database schema at the INFO logging level. Now, ClaimCenter logs differences at the WARN logging level.
- PL-20200 Guidewire has implemented optimization on the database upgrade. This optimization improves 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.
- PL-21357 The Upgrade Info page and download now provide more consistent information about the steps that are being performed during an automated database upgrade.
- PL-21387 Corrected an issue that caused a null-pointer exception in version triggers during a database upgrade.
- PL-21452 Corrected a issue that caused Document Management file downloads to fail if running over SSL in Internet Explorer 8 or earlier.

**DBChecks – Consistency/Integrity, Staging Tables, LoaderCallbacks, Constraints - Core**

- CLM-18208 Fixed an issue in which certain version checks running prior to upgrade would throw the following exception: ClassCastException: java.lang.Long cannot be cast to java.lang.Integer.

**Document Management**

- PL-18278 Guidewire added a new configuration parameter, AllowActiveXAutoInstall. This parameter controls whether ClaimCenter 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, ClaimCenter 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 *System Administration Guide*.
- 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).
- PL-19958 Guidewire template documents using 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 (white list) to 2.0.52/2.1.52. Customers that manually deploy the controls will need to do so again.
- PL-20275 Guidewire Document Assistant no longer adds extra section breaks to generated Microsoft Word documents.
- 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.
- PL-20549 Fixed a race condition in the initialization of hidden frames for ClaimCenter that could cause the supporting JScripts for the Document Assistant ActiveX control TemplateRunner to not be downloaded.

**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 ClaimCenter 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-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-18068	Fixed a problem with verification of typelist typecodes on server startup. Typelist typecodes that match regardless of case fail validation. For example, the following two typecodes would fail validation: <ul style="list-style-type: none"> <li>• Foobar and FooBar</li> </ul>
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.
<b>File Export</b>	
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.
<b>Financials – Bulk Invoices</b>	
CLM-19739	Fixed an issue in which attempting to get the associated payment for an invoice item of a voided or stopped bulk invoice would throw an exception.
CLM-19737	Fixed an issue in which bulk invoices can either get stuck during submission processing or advance, even though one or more invoice items have not finished processing.  If a bulk invoice unexpectedly stays in Pending Item Validation status for a long time, an administrator must run the Bulk Invoice Submission batch process again. If this situation occurs often, please consult Guidewire Support, since this condition might indicate a misconfiguration.
<b>Financials – Checks, Payments, Deductibles</b>	
CLM-19738	Added a new configuration parameter to the config.xml file, DeleteCompletedApprovalActivitiesOnEdit, which defaults to true. This default setting preserves base configuration functionality that deletes all approval activities on the check set when resubmitting a check or bulk invoice or when invalidating a bulk invoice. If this parameter is set to false, any existing completed approval activities are preserved when resubmitting a check or bulk invoice or when invalidating a bulk invoice. Open approval activities will be deleted regardless of the setting.
<b>Financials – Reinsurance</b>	
CLM-19575	Removed a database consistency check that incorrectly prevented reinsurance agreements from having negative total ceded reserves. Negative total ceded reserves are allowed.
<b>Financials – Subrogation</b>	
CLM-20807	Fixed an issue in which a null pointer exception might be thrown when using a property for which intermediate entities are null. The problem occurred when restoring an archived claim if the SubroPaymentSchedule.SubroInstallmentAmount field was populated. Now if an intermediate entity is null, using a property that depends on the intermediate entity returns null rather than causing an NPE. For example, using SubroAdverseParty.Claim when SubrogationSummary is null no longer causes an NPE, but rather returns null.
<b>Global Cache</b>	
PL-20332	A new configuration parameter, GlobalCacheDetailedStats, determines whether to collect detailed statistics for the global cache. Detailed statistics are data that ClaimCenter 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.
<b>Gosu</b>	
PL-20961	In Gosu you can now subclass an inner class that a supertype declares.
<b>Integration CM – Core: Link, Related Contacts, Match, Retrieve, Search, Update</b>	
CLM-20311	An issue where ClaimCenter was improperly enforcing an old ContactManager requirement has been fixed.
<b>Messaging</b>	
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 Java.MessageDestination level of the display key namespace.

PL-13887	<b>IMPORTANT:</b> 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.  The <code>IMessagingToolsAPI</code> has a new method to retrieve the status of a messaging destination. The method signature is:  <code>getDestinationStatus(destID : int) : String</code>
<b>Plugins</b>	
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> .
<b>Printing</b>	
CLM-17729	Fixed an issue with printing second-level list views. They are now printed correctly.
<b>Search – Claim, Activities</b>	
CLM-14321	The <code>ClaimContactCell</code> widget now has the option to hide the Search menu.
<b>Staging Tables</b>	
PL-20303	ClaimCenter provides two new consistency checks: “One-to-one non-null check” and “Edge foreign key non-null check”.
<b>Studio IDE</b>	
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 <b>Section Inclusion Limit</b> in Tools → Verification Options → PCF Verification Options.
PL-21296	Modified the Studio PCF Editor to make the PCF Toolbox show 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.
<b>Web – UI/Runtime</b>	
PL-20150	Modified ClaimCenter event-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, the second event will no longer be processed. The result is that the associated <code>onChange</code> handler will not be invoked to avoid errors.
<b>Web Services</b>	
PL-21158	The Guidewire Profiler previously did not profile WS-I web services. Now whenever you select a WS-I webservice to profile, ClaimCenter provides profile information.
<b>Web Services - WSI</b>	
PL-18729	It is now possible to invoke local web services using the <code>wsdl.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.
<b>XML Element - XSD types</b>	
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 ClaimCenter.

- “ClaimCenter Known Issues” on page 244
- “Studio/Platform Issues” on page 245

**Note:** For maintenance releases, Guidewire often defers fixing configuration issues if they require merging files during the upgrade. Workarounds to some of these issues are listed in the following sections. The goal of this policy is to make upgrades as straightforward as possible.

## ClaimCenter Known Issues

### Implementing Policy Refresh

Before implementing policy refresh, Guidewire strongly recommends that you contact Customer Support for guidance on this implementation.

### New Claim wizard not saving selected vehicle in some cases (CLM-13973)

**Issue** – The second step of the New Claim wizard is not saving your vehicle selection when you leave the screen and then come back. For example, you choose a vehicle and then click the **Policy Details** link, and when you click **Basic Info** on the left to return to this screen, the vehicle is no longer selected.

**Workaround** – Reselect the vehicle when you return to the wizard screen.

### Upgrade from ClaimCenter 6 to ClaimCenter 7 does not work for SQL Server if MigrateToLargeIDsAndDatetime2 is set to true (CLM-17739)

**Issue** – During upgrade with SQL Server as the database, the `cc_tmppagglimitrpt` table fails to be upgraded if it is not empty and `MigrateToLargeIDsAndDatetime2` is true.

**Workaround** – If there is data in the table `cc_tmppagglimitrpt`, truncate the table before beginning the upgrade by running the following SQL command:

```
DELETE FROM cc_tmppagglimitrpt
```

### ABContactPlugin now requires ContactManager 7.0.1 or later (CLM-17990)

**Issue** – The class `gw.plugin.addressbook.ab700.PluginRetrieveOperation`, which is used by `ABContactPlugin`, has a method that requires the `ABContactAPI` web service in ContactManager 7.0.1 or later. This method, `retrieveReplacementContact`, will cause errors if your ClaimCenter application continues to be integrated with ContactManager 7.0.0. Guidewire recommends that you upgrade to ContactManager 7.0.1 or later. If you cannot upgrade ContactManager, see the workaround for this issue.

**Workaround** – Copy `ClaimCenter/modules/cc/gsrc/gw/plugin/addressbook/ab700/PluginRetrieveOperation.gs` to `ClaimCenter/modules/configuration/gsrc/gw/plugin/addressbook/ab700/PluginRetrieveOperation.gs`. Edit the copied file and remove the annotation `@ReadOnly`, the `retrieveReplacementContact` method, and all calls to the `retrieveReplacementContact` method. Then save the file.

### Cannot upgrade database from ClaimCenter 5.0.8 to ClaimCenter 7.0.3 (CLM-20484)

**Issue** – Attempting to upgrade the database from ClaimCenter 5.0.8 to ClaimCenter 7.0.3 produces an error and fails.

**Workaround** – For Commercial Auto, ClaimCenter 7.0.0 and earlier had coverage types of Auto Death Benefit, TypeKey.CoverageType.BADeathCov, and Auto Disability Benefit, TypeKey.CoverageType.BADisabilityCov. The equivalents for PolicyCenter 7.0.3 and later are one coverage type of Death and Disability with two coverage terms, Death Benefit and Disability Benefit. ClaimCenter Commercial Auto coverages have been changed to match those of PolicyCenter. In Guidewire Studio, the new typekeys in Localizations → Typelist Localizations are CoverageType.CADeathDisabilityCov, CovTermPattern.DeathBenefit, and CovTermPattern.DisabilityBenefit.

If you use PolicyCenter and have Commercial Auto claims, this change might affect upgrading from ClaimCenter 7.0.1 or earlier to ClaimCenter 7.0.3. If your PolicyCenter and ClaimCenter configurations are not already in sync, you will have the opportunity to make these changes during the merge step of the configuration upgrade process. If you do make these changes, you must use SQL to update claims that use the Auto Death Benefit and Auto Disability Benefit typekeys to point to the new single typekey. Additionally, you must insert a CovTerm row to indicate if the coverage includes the death benefit or the disability benefit.

## Studio/Platform 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, Internet Explorer 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 an Internet Explorer 9 workaround that reduces these issues. In Internet Explorer 9, you can try to change the use of software or hardware rendering by toggling the Accelerated Graphics option on the Advanced tab of the Internet Options dialog.

### Studio Rules do not use logical capitalization for root object's name (PL-10740)

**Issue** – Rule set root objects are named with first letter lower-cased, which can look strange for variables like aBContact.

**Workaround** – Use the variable name as provided by Guidewire Studio.

### 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** – Entity localization table names have a shorter, 16-character limit than other entity tables. If the localization table name exceeds the 16-character limit, the error message indicates incorrectly that 18 characters are allowed.

**Workaround** – Ensure that the localization `tableName` property specified in the entity extension file is less than 16 characters.

#### [US-Locations.txt file with the US geodata from Great Data 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 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 `ClaimCenter/bin/gwpc regen-java-api` command, ClaimCenter creates a `ClaimCenter/java-api/lib` directory and puts JAR files for Java development in this directory. However, ClaimCenter 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:

`ClaimCenter/admin/lib`

Copy them into the following directory:

`ClaimCenter/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, ClaimCenter throws `ParseResultsException`. This is the intended behavior.

**Workaround** – Restart the workflow engine. To do so:

1. Log into ClaimCenter using an administrative account.
2. Access Internal Tools → Reload.
3. Click Reload Workflow Engine.

# Guidewire ClaimCenter 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:** 23-June-2016

Copyright © 2001-2016 Guidewire Software, Inc.

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, Guidewire Spotlight, Gosu, Adapt and succeed, 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:

- “Release Information” on page 248
- “Installing This Release” on page 248
- “Support” on page 248
- “Issues and Major Changes” on page 248
- “Improvements and General Issues” on page 250
- “Known Issues and Limitations” on page 259

## Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

### Version Number

This release of Guidewire ClaimCenter is 7.0.2.

## Installing This Release

Refer to the following:

- The *ClaimCenter Installation Guide* for general installation information
- Prior ClaimCenter 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 might affect your installation.

- “Base PCF File Changes” on page 249
- “Rules Changes” on page 249
- “Changes in This Release Provided in Upgrade Diff Report” on page 249

### Manual Checks Are Not Allowed to Be Negative (CLM-18333)

There is a new configuration parameter, `AllowNegativeManualChecks`, that indicates whether ClaimCenter allows negative manual checks. The default value is `false`. This behavior is a change from ClaimCenter 7.0.0 and 7.0.1, which in the base configuration allow manual checks to be negative. To enable ClaimCenter to allow negative manual checks, edit `config.xml`, add the configuration parameter `AllowNegativeManualChecks`, and set it to `true`.

## Base PCF File Changes

All links below require the `ReleaseNotes_files` directory on your local disk.

### **ClaimCenter release 7.0.1 to 7.0.2**

- To view a report of the changes in the base PCF files in the `modules/cc` directory, *[click here](#)*.
- To view a report of the changes in the base PCF files in the `modules/p1` directory, *[click here](#)*.

## Rules Changes

### **ClaimCenter release 7.0.1 to 7.0.2**

- There are no changes to the base rules in the `modules/cc` directory.

## 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

The following are the primary improvements and issues corrected in this release:

ID	Description
<b>Activities</b>	
PL-19637	Fixed an issue that caused the Activity Escalation work queue writer to create duplicate work items.
<b>Administration</b>	
CLM-45	Added the display keys Java.AuthorityLimitProfiles.CustomProfile.Name = Custom and Java.AuthorityLimitProfiles.CustomProfile.Description = Custom Profile to replace two hard-coded strings in AuthorityLimitProfile.
<b>Archiving</b>	
PL-18324	Fixed an issue with claim archiving that caused a loss of data if a database upgrade occurred between the archiving of the claim and restoration of the claim under certain circumstances.
CLM-17654	When non-fatal issues occur during upgrade of an archived claim, a note is created and attached to the claim. You can now control whether this note is created at all, the subject and body of the note, and the technical message added to the server logs, as follows: <ul style="list-style-type: none"> <li>A configuration parameter was added to config.xml that controls whether or not the note is created. The parameter, CreateNoteWithArchiveUpgradeIssues, defaults to true.</li> <li>The subject and body of the note are controlled by display keys. The body message is in Java.Archive.API.Warn.UpgradeIssues.NoteBody, and the subject is in Java.Archive.API.Warn.UpgradeIssues. The list of issues is passed to the NoteBody in case you want to display the list.</li> <li>The technical message is now added to the server logs to ensure that the information is not lost. That display key is Java.Archive.API.Warn.UpgradeIssues.LogMessage. It takes two arguments, the claim number and a string containing the list of upgrade issues.</li> </ul>
<b>Assignment, Email, Notes, Documents, Question Sets</b>	
CLM-17488	The method DocumentProduction.adjustDocumentName was moved into a new gw.document.DocumentEnhancement class, and a call to the method DocumentEnhancement.preSynchronousStore was added in its original place. The method adjustDocumentName is used for ensuring the uniqueness of a document name in your Document Management System (DMS) and might need to be configured for each DMS. The method was previously not visible because it was a private Gosu method in a class marked @ReadOnly.  In particular, you might need to override this method if your DMS checks for the existence of a document by something other than document.Name, such as docUID. If docUID is used to test for the existence of a document and the document is present, calling the default implementation of adjustDocumentName results in an infinite loop. In this case, you need to override adjustDocumentName with your own code for naming the document uniquely.
<b>Batch Process</b>	
PL-18232	It is now possible to extend any WorkQueue subclass exposed in ClaimCenter.
PL-18404	Removed the BatchServer configuration parameter from config.xml. Instead, use the JVM argument or registry element isBatchServer to set the batch server.  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.  Also, added the promoteToBatch operation to the Cluster Management Bean.
PL-18637	The default ClaimCenter 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.  <pre>public final void setChunkingByID(IQueryResult queryResult, int chunkSize)</pre>
<b>Claim File - Associations, Calendar, Claim Status, Evaluation, Exposures, History, Incidents, Loss Details, Matters, Negotiation, SIU, Summary</b>	

---

CLM-17838	<p>For Commercial Auto, ClaimCenter 7.0.0 had coverage types of Auto Death Benefit (<code>CoverageType.BADeathCov</code>) and Auto Disability Benefit (<code>CoverageType.BADisabilityCov</code>). The equivalents for PolicyCenter 7.0.3 and later are one coverage type of Death and Disability with two coverage terms, Death benefit and Disability benefit. ClaimCenter Commercial Auto coverages have been changed to match those of PolicyCenter. In Studio, the new typekeys in <b>Localizations -&gt; TypeList Localizations</b> are <code>CoverageType.CADeathDisabilityCov</code>, <code>CovTermPattern.DeathBenefit</code>, and <code>CovTermPattern.DisabilityBenefit</code>.</p> <p>If you use PolicyCenter and have Commercial Auto claims, this change might affect upgrading from ClaimCenter 7.0.0 or 7.0.1 to ClaimCenter 7.0.2. If your PolicyCenter and ClaimCenter configurations are not already in sync, you will have the opportunity to make these changes during the merge step of the configuration upgrade process. If you do make these changes, you must use SQL to update claims that use the Auto Death Benefit and Auto Disability Benefit typekeys to point to the new single typekey. Additionally, you must insert a CovTerm row to indicate if the coverage includes the death benefit or the disability benefit.</p>
-----------	---

**Claim Metrics**

CLM-17778	<p>Previously, it was not possible to configure the claim metrics update process to run on some claims but not on others. This fix provides a way to do that.</p> <p>The interface delegate <code>ClaimUpdateMetricsMethods.java</code> was added to the <code>Claim</code> entity and implemented in Gosu in <code>gw.claim.metric.ClaimUpdateMetricsMethodImpl1</code>. In the base configuration, the delegate calls <code>ClaimHealthUpdater.updateClaim</code>, causing the claim metrics update process to run on all claims. To configure which claims the claim metrics process runs on, you can alter the existing <code>ClaimUpdateMetricsMethodImpl1</code> class.</p>
-----------	---

**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 two or more 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	<p>Improved the usability of the <b>Cluster Info</b> screen on the <b>Server Tools</b> tab. The screen now shows Server ID and host name information. Guidewire also changed the menu slightly.</p>

**Cognos Integration**

PL-19226	<p>The <code>cc cognos</code> command now builds an application-specific Cognos integration ZIP file, with a name of the form <code>cc-cognos-reporting-integration.zip</code>. Each zip file contains application-specific integration scripts, <code>cc-config</code> and <code>cc-import</code>. These scripts no longer require the namespace/application parameter.</p>
----------	--

**Command Line Tools**

PL-13663	<p>Removed hard-coded Java heap settings that could cause out-of-memory issues when running certain tools.</p>
PL-18225	<p>Added WSI document literal web services support to Administration command-line tools.</p>
PL-18856	<p>Fixed an issue that involved regenerating the Data Dictionary. Running <code>gwcc regen-dictionary</code> did not actually produce the Data Dictionary files.</p>
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 <code>verify-all</code>. For example:</p> <pre>gwcc 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>

**Note**

- If you do not include this option, the dictionary generation tool behaves as before.
- Use only positive integer values for the `maxSPVInclusions` property.

**Command Line Tools, Toolkit, Data Dictionary, External Entities, CC Batch Process, Financials - Summary, Search, TransactionsLV, ChecksLV, Rpts**

CLM-19248	The batch process that produces Dashboard Statistics was throwing an error if the datatype MoneyDataType was set in datatype.xml to have an appscale of 0. The problem has been fixed.
<b>Configuration Upgrade</b>	
PL-12804	Fixed an issue in which the upgrade tool did not handle rules upgrade properly.
<b>Consistency Checker</b>	
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.
<b>Contact Domain</b>	
PL-18245	Fixed an issue with IgnoreProperty used in contact-sync-config.xml that caused it to take effect 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.
<b>Contacts, Roles, SOAP APIs, Plugins, Rules - Core</b>	
CLM-17993	<p>Fixed a problem with the &lt;IgnoreProperty&gt; element in contact-sync-config.xml. The property defined in this element was being ignored during Address Book synchronization for all contact subtypes rather than only for the contact subtype for which the element was defined. The element now applies to the Contact subtype for which it is defined and to any subtypes of that element unless appliesToSubtypes = false is specified.</p> <p>In addition, note that the behavior of &lt;IgnoreProperty&gt; has changed in ClaimCenter 7.0. It does not control the transmission of updates of the designated property in ClaimCenter to ContactManager as it did in prior releases. In ClaimCenter 7.0 and later, the ContactIntegrationXMLMapper class determines which Contact properties are sent between the applications. The &lt;IgnoreProperty&gt; element in contact-sync-config.xml still determines if changes to the designated property are copied into a local ClaimCenter Contact. It also continues to determine if the property is used to generate the Address Book fingerprint to determine sync status.</p> <p>You must take an extra step to get the same behavior as in prior releases, in which changes to the property that is defined in &lt;IgnoreProperty&gt; do not get propagated back to ContactManager. You must edit ContactIntegrationXMLMapper and change the code so that the property is not mapped to XML and sent to ContactManager.</p>
<b>Database</b>	
PL-1541	ClaimCenter now explicitly names primary key constraints by using the following conventions: <ul style="list-style-type: none"> <li>• Oracle – Primary key constraint on the table PX_CONTACT will be named PK_CONTACT.</li> <li>• All other databases – Primary key constraint on the table PX_CONTACT will be named PX_CONTACT_PK.</li> </ul>
PL-16199	One-to-one relationships are required to 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 requirement. In those cases, Guidewire now creates a consistency check to verify the integrity of the data.
PL-17234	ClaimCenter no longer creates shadow tables—tables whose names start with XXt_ and XXt_—for new databases. If you start the application server in dev mode (JVM parameter -Dgw.server.mode=dev), ClaimCenter now creates shadow tables only if you set the server.running.tests system property to true, either 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	ClaimCenter 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 <b>Upgrade Info</b> screen on the <b>Server Tools</b> tab.
PL-18637	The default ClaimCenter 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 behavior can cause problems. Now you can use the following method to set the query retrieve chunk size.
<code>public final void setChunkingByID(IQueryResult queryResult, int chunkSize)</code>	

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	<p>Fixed an issue on SQL Server that caused a stack trace overflow if both of the following were true:</p> <ul style="list-style-type: none"><li>• There was a connection problem during application start-up.</li><li>• Configuration parameter <code>MigrateToLargeIDsAndDatetime2</code> was set to <code>false</code>.</li></ul>

#### 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:  <a href="http://bfo.com/viewtext.jsp?url=products/pdf/docs/CHANGELOG.txt&amp;title=Big+Faceless+PDF+Library+Changelog">http://bfo.com/viewtext.jsp?url=products/pdf/docs/CHANGELOG.txt&amp;title=Big+Faceless+PDF+Library+Changelog</a>
PL-18704	Microsoft Word field forms—text fill-in fields—are now restored correctly while generating a document from a Microsoft Word Template.

#### Entities/Metadata

PL-14982	Modified the Studio Tools → Verify... feature so that it now reports errors and 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"><li>• Entities that do not have a class</li><li>• Entities that are extended with a delegate or Java interface that the compile-time backing class does not extend</li></ul> 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.
PL-18469	Added a new attribute called <code>typelistTableName</code> to entity types. Use this attribute for non-final entities to specify the name of the corresponding subtype typelist table. If not specified, ClaimCenter 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, ClaimCenter prevents you from committing empty <code>oneToOne</code> columns that do not allow nulls.
PL-20217	There was a problem 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 when 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.

#### Financials - Bulk Invoices

CLM-19245	On a bulk invoice set to split its amount evenly across all line items, ClaimCenter calculated the split amount to two decimal digits regardless of the bulk invoice's currency or configuration of the money data type. This hard-coding has been removed, and ClaimCenter now uses the scale of the bulk invoice currency.
-----------	--

#### Financials - Checks, Payments, Deductibles

CLM-15777	When the configuration parameter <code>CheckAuthorityLimits</code> in <code>config.xml</code> is set to <code>false</code> , a user who has check creation permissions but not authority limits is supposed to be able to create checks. This was not happening. This bug has been fixed by updating user permissions to ignore authority limits if the <code>CheckAuthorityLimits</code> configuration parameter is set to <code>false</code> . With this change, the quick menu items on the <b>Financials Summary</b> screen can be enabled if the user has permissions to use them.
CLM-17429	On the <b>Change Check Recurrence</b> screen, sometimes a change to the scheduled send date of a check could not be saved if a validation warning was displayed. This screen now uses a more standard approach in committing and canceling changes, ensuring that it handles validation warnings appropriately.
CLM-17430	Fixed a bug that could occur when editing a future-dated payment. The system might delete the payment's existing zeroing reserve and not check to see if a new zeroing reserve needs to be created based on the payment's new amount. Modified the <code>zeroReserves</code> method in the <code>CCPReupdateAbstractHandlerImpl</code> Gosu class to handle this situation. The changes to this Gosu file must be properly merged into your configuration to fix this bug.
CLM-18064	Fixed a bug that could occur while editing or recoding a check if the <code>MulticurrencyDisplayMode</code> parameter was changed from <code>SINGLE</code> to <code>MULTIPLE</code> . The bug could occur if a check created under the first setting was then edited or recoded as a split payment while under the second setting. Exchange rate data on a transaction could have been set up differently depending on the <code>MulticurrencyDisplayMode</code> configuration parameter. The enforcement of payment exchange rates on a <b>Check</b> entity was fixed to allow both kinds of representation on the same check.

#### Financials - Recovery, Recovery Reserves

CLM-17880	There was an error occurring when retiring a <code>CostCategory</code> that would otherwise be available as a valid <code>CostCategory</code> on the <b>New Recovery</b> screen. Fixed code that was calling <code>remove</code> on a collection of <code>CostCategory</code> entities while iterating over it. That code now uses a copy of the collection. Also made changes to ensure that a list of possible exposures used in the user interface was properly initialized before use.
CLM-18057	Added methods to make creating recovery transactions easier in Gosu. Specifically, created the following methods: <ul style="list-style-type: none"> <li>• <code>Claim.newRecoverySet</code></li> <li>• <code>RecoverySet.newRecovery</code></li> <li>• <code>RecoverySet.prepareForCommit</code></li> </ul>
CLM-18195	Made the <code>OffsetRecoveryReserve</code> property on <code>Recovery</code> visible in Gosu.

#### Financials - Reinsurance

CLM-17442	There was a bug concerning how ClaimCenter handles reinsurance agreements with duplicate numbers. Changed the reinsurance plugin to process the external definition of internal <code>RIAgreement</code> objects more robustly when they have duplicate agreement numbers.
CLM-17553	Loosened the restrictions around creating Reinsurance (RI) Transactions to issue warnings only when RI Transaction amounts seem out of bounds.  Previously, the system would throw exceptions or validation errors when RI Transactions resulted in negative or relatively large values. Relatively large values were RI Transactions that were greater than their corresponding financial transactions. For example, RI Ceded Reserves were greater than Committed Open Reserves, or RI Recoverables were greater than Committed Payments.  As before, the system directly warns a reinsurance manager when the values being created are out of bounds. If regular financial transactions result in out of bounds RI transactions, the system creates an activity for the reinsurance manager to examine the claim.
CLM-18104	Changed the reinsurance plugin so that it does not attempt synchronization for removed exposures. Made the changes in both <code>gw.plugin.policy.reinsurance.impl.ReinsuranceDemoPlugin</code> and <code>gw.plugin.policy.reinsurance.pc700.PCReinsurancePlugin</code> .
CLM-18161	There was a problem with removing coverage from an exposure with reinsurance. After removing the coverage, the exposure was supposed to have been unlinked from the reinsurance agreement group, but it was not. This bug has been fixed. Updated <code>gw.plugin.policy.reinsurance.impl.ReinsurancePluginHelper</code> to remove an exposure from an RI Agreement group if it was no longer valid to have RI Agreements associated with the exposure.
CLM-18163	Changed when Workers Comp exposures are automatically associated with a coverage. Previously this association was performed in a rule, and now it is part of the creation of the exposure. This change ensures that the exposure is sufficiently complete at the beginning of the commit cycle to allow processes that depend on the exposure to use it during commit. For example, if there is a reinsurance agreement covering the exposure, it is now committed with the exposure.

CLM-18185	It is valid for a Workers Comp claim to have a recovery limit for a reinsurance agreement that is null, indicating an unlimited amount of total risk. To support this scenario, changed RIAGreement to allow RecoveryLimit to be nullable.
CLM-18211	There was a bug that allowed a user to add an exposure to a reinsurance agreement group that was not editable, like reinsurance agreements obtained from external systems. Changed ReinsuranceExposures.pcf to display only editable RIAGroup objects when associating one with an exposure.
CLM-18245	Changed the reinsurance plugin gw.plugin.policy.reinsurance.pc700.PCReinsurancePlugin. This plugin now unlinks an exposure from an RIAGroup if the reinsurance system returns an empty or missing collection of RIAGreement objects for the exposure.
CLM-19532	A problem was reported with setting ceded reserves or recoverables adjustments on an agreement. If the agreement was associated with three reserve lines, or any number that did not divide evenly into 1 with a finite number of digits, an arithmetic exception could be thrown. This problem has been fixed by correctly rounding the proportions.

**Financials - ReserveLines, TAccounts, Financials Calculations**

CLM-13698	Deprecated the following methods in FinancialsLibrary: <ul style="list-style-type: none"> <li>• getAvailableReservesForCostCategory(Transaction t)</li> <li>• getAvailableReservesForCostCategory(Transaction t, CostCategory costCategory)</li> <li>• getPendingReservesForCostCategory(Transaction t)</li> <li>• getPendingReservesForCostCategory(Transaction t, CostCategory costCategory)</li> </ul> Deprecated the following methods in FinancialsCalculationUtil: <ul style="list-style-type: none"> <li>• getAvailableReservesForCostCategory(Transaction t)</li> <li>• getAvailableReservesForCostCategory(Transaction t, CostCategory costCategory)</li> <li>• getPendingReservesForCostCategory(Transaction t)</li> <li>• getPendingReservesForCostCategory(Transaction t, CostCategory costCategory)</li> </ul> A replacement expression is included in the Javadoc for each method.
-----------	--

**Financials - SOAP APIs, Plugins, Rules**

CLM-18191	For the Financials configuration parameter in config.xml, the possible value of 3—subledger—has been removed. Now only view and entry are valid values. The subledger option has been obsolete and unused for at least three major releases.
-----------	--

**Geocoding/Proximity Search**

PL-1461	Two parameters have been added to the plugin registry for the Geocode plugin. The geocodeDirectionsCulture parameter specifies the locale for geocoded addresses and routing instructions returned from a geocoding and routing service. For example, use the locale code ja-JP for addresses and instructions for Japan. The imageryCulture parameter specifies the language for map imagery. For example, use the language code ja 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 DisableIndexFastFullScanForProximitySearch and DisableHashJoinForProximitySearch control the workaround. The default value for these parameters is false. These parameters have no effect on databases other than Oracle.

**Global Cache**

PL-18322	Fixed an issue with method ProximitySearchQueryUtils.filterWithinRadiusLatLong 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.
----------	--

**I18N, Language Pack, Localized Modules**

CLM-17971	Fixed a bug with a hard-coded string, "Action Failed", that could not be translated. This string is now localized in the display key Java.Archiving.Schedule.Failed.
-----------	--

**Integration PC, Policy Refresh**

CLM-18170	Fixed an error with the wrong exception being thrown when a policy wasn't found in PC after clicking Policy Refresh.
-----------	--

**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 Batch Process Info screen on the Server Tools tab to enable you to download detailed records for any particular batch process for a specific date range. Clicking Download now opens a screen in which you can specify the date range to download for a given batch process.

<b>Localization</b>	
PL-18490	Modified <code>localization.xml</code> to let the <code>thousandsSymbol</code> attribute of the <code>&lt;NumberFormat&gt;</code> element type accept a non-breaking space character as the thousands separator. For example:  <code>&lt;NumberFormat thousandsSymbol="&amp;#160;&gt;</code>
<b>Logging</b>	
PL-14722	Added the following new configuration parameters that configure application logging: <ul style="list-style-type: none"><li>• <code>LoggerCategorySource</code></li><li>• <code>LoggersShowLog4j</code></li></ul>
PL-14764	Added the <code>UserInterface.Performance</code> logging category for tracing and debugging issues in the user interface. You can configure the logging category to include the ID of the user.
PL-18745	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> :  <code>%-&lt;LL&gt;. &lt;HH&gt;X{user   userName   userID}</code> The parameters have the following meanings: <ul style="list-style-type: none"><li>• <code>&lt;LL&gt;</code> – The minimum size of the field. If the actual value is shorter, the user name gets padded with spaces on the right.</li><li>• <code>&lt;HH&gt;</code> – The maximum size of the field. If the actual value is longer, the user name gets truncated from the left.</li><li>• <code>user</code> – Prints the user's internal numeric ID number, such as 4231341234. This parameter was available in previous releases and remains unchanged for compatibility.</li><li>• <code>userName</code> – Prints the user's real-world name, such as John Smith.</li><li>• <code>userID</code> – Prints the user's username in the system, such as <code>jsmith</code>.</li></ul> For example:  <code>%-16.16X{userName}</code>
PL-20907	Guidewire updated the logging API between the following releases: <ul style="list-style-type: none"><li>• BillingCenter 7.0.1 and 7.0.2</li><li>• ClaimCenter 7.0.1 and 7.0.2</li><li>• PolicyCenter 7.0.3 and 7.0.4</li></ul> Guidewire has created the following Knowledge Base article that describes the changes made to the logging API:  <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>  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.
<b>Messaging</b>	
PL-18678	The <code>MessageSenderRunnable.run</code> method now includes the <code>messageID</code> in the log. The log entry will look similar to the following:  <code>MW.MessageSenderRunnable.run (dest destination_id): Entering run() for messageId message_id</code>
PL-19136	Fixed an issue in which a race condition in a non-clustered server could lead to inconsistent results or runtime exceptions or both when modifying entities within a message transport.
PL-19149	After locking out a user who has reached the maximum number of allowed login failures, ClaimCenter no longer ignores generated events. Instead, ClaimCenter executes the transaction with a service token having a super user. As a result, ContactManager now generates UserChanged events properly.
<b>Miscellaneous</b>	
CLM-17387	Fixed a displaykey issue where Matter might incorrectly be referred to as Exposure.
CLM-18269	The populator for claim access staging tables was changed to use the public ID prefix configured in the <code>config.xml</code> parameter <code>publicIDPrefix</code> . Previously, the populator was incorrectly hard-coded to use the default cc prefix.

CLM-18492	<p>In ClaimCenter 6.0.6, a problem was fixed that was resulting in duplicate query predicates when visiting the vacation tab to view claims or exposures (CLM-15037). Unfortunately, this fix introduced a performance problem and a bug in the code that checks whether to show the vacation tab. The vacation tab started showing up even when the current user did not have any open claims or exposures to see on the tab. The query that was used to verify visibility was not only wrong, but it was expensive.</p> <p>Both fixes were ported forward to ClaimCenter 7.0.2. If you are upgrading from 6.0.6 and you used our workaround, you need to remove <code>GWPublicClaimVacationViewFinderEnhancement.gsx</code> and <code>GWPublicExposureVacationViewFinderEnhancement.gsx</code>. The functionality of those files has been moved into the core product.</p>
PL-3760	<p>Guidewire has made the following modification to <code>config.xml</code>:</p> <p>Removed element <code>&lt;security sessiontimeoutsecs="10800"/&gt;</code></p> <ul style="list-style-type: none"> <li>Added configuration parameter <code>&lt;param name="SessionTimeoutSecs" value="10800"/&gt;</code> as its replacement</li> </ul> <p>ClaimCenter performs an automatic upgrade of <code>config.xml</code> for this change.</p>
PL-15462	<p>Guidewire applications provide several configuration parameters that accept a comma-separated list of values.</p> <p>Now, ClaimCenter parses a comma-separated list of values according to CSV format and trims each value automatically. This change enables you to use spaces, tabs, and new lines for more readability while specifying such a value.</p> <p>For example:</p> <ul style="list-style-type: none"> <li>A value that before the fix had to be without spaces – AUTO,PR,GL,TRAV</li> <li>The same value after the fix can use spaces – AUTO, PR, GL, TRAV</li> </ul> <p>The following is a more technical description of the format:</p> <ul style="list-style-type: none"> <li>Fields must be separated with commas.</li> <li>Leading and trailing spaces are ignored unless the field is delimited with double-quotes. In that case, the white spaces are preserved.</li> <li>Embedded comma – You must delimit the field with double-quotes, with the comma inside the double quotes.</li> <li>Embedded double-quotes – You must double embedded double-quote characters, and you must delimit the field with double-quotes.</li> <li>Embedded line-breaks – You must surround the field with double-quotes.</li> <li>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.</li> </ul>
PL-15914	Guidewire has added Save and Cancel buttons on the Server Tools → Management Beans → Guidewire Managed Bean Properties screen 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.
PL-18245	Fixed an issue with <code>IgnoreProperty</code> used in <code>contact-sync-config.xml</code> that caused it to take effect 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.
PL-19311	Guidewire has upgraded the Joda-Time third-party library version to 2.0.
<b>Page Configuration</b>	
PL-15548	Printing a second-level list view no longer results in missing items or items to print more than once.
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.
<b>Persistence</b>	
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.
<b>PL Services</b>	
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 two or more batch servers in the cluster if the cluster was not completely formed yet.</p> <p>This change uses the underlying database to discover and keep track of the batch server that is currently active in the cluster.</p>

PL-18232	It is now possible to extend any WorkQueue subclass exposed in ClaimCenter.
<b>Policy, LOB configuration, Upgrade - Database or Configuration</b>	
CLM-17349	When an error occurs during Line of Business (LOB) typelist validation, an additional check for type-codes with duplicate names is run. Any duplicates found are included in the error message.
<b>Queries</b>	
PL-17541	Modified GWDBFunctionEnhancement.gsx and added support for DBFunction calls of MIN and MAX on java.util.Date.
PL-19590	Fixed an issue that caused an error in the ClaimCenter user interface by a query that contained a column definition twice.
<b>Rules Editor</b>	
PL-19466	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 and moves the top rule down rather than overwriting it.
PL-19510	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.
<b>Studio</b>	
PL-19568	Guidewire Studio now supports 64-bit Java Virtual Machines (JVMs).
<b>Studio IDE</b>	
PL-13352	Fixed an issue in which Studio ignored the formatting settings for String literals set in Studio Tools → Options → Colors and Fonts.
PL-15219	Fixed an issue with the TemplateInputWidget PCF file in which autocomplete inserted a value that appeared to be correct, but failed validation.
PL-15887	Fixed an incorrect message that Studio showed in the Verify Result pane if verifying a valid entity name in the Entity Names editor.
PL-18203	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.
PL-18671	Guidewire has modified Studio behavior in regards to readonly mode as follows: <ul style="list-style-type: none"><li>• 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 shows a modal message box indicating the reasons why it is in readonly mode.</li><li>• Studio disables the Save button any time that it is in readonly mode.</li><li>• Studio changes the Save button tooltip in readonly mode to show the reason that save is not active in this mode. This message is the same one that Studio shows if you click the padlock icon on the status bar.</li></ul>
PL-18905	Fixed an issue in which it was possible to edit configuration files even if Studio was in readonly mode.
<b>Suite</b>	
PC-16371	PolicyCenter provides the ability to export the PolicyCenter product model to the ClaimCenter Line of Business model. This process was not working properly for coverages created in ClaimCenter. This export process is now fixed. It reads in all existing ClaimCenter Line of Business elements that were created natively in ClaimCenter, adds the PolicyCenter model elements to them, and then exports everything back to ClaimCenter.
<b>Upgrade: Database or Configuration</b>	
CLM-18018	The Add Button widget has a default toCreateAndAdd method that was throwing an error if the user attempted to add a custom entity. This method has been fixed by PL-18308, which enabled adding multiple entries in list views for custom entity types.
<b>Utilities</b>	

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 <a href="http://jakarta.apache.org/oro/">http://jakarta.apache.org/oro/</a> 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.substitute(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.
<b>Web</b>	
PL-15548	Printing a second-level list view no longer results in missing items, nor does it cause 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.
<b>Web Services</b>	
PL-18361	ClaimCenter now provides the ability to implement a pre-existing WSDL in Gosu for a WSI web service.
PL-18450	ClaimCenter 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 the logging category <code>XML.Request</code> for WSI web services to log each request. Each request will be logged at the DEBUG level. This logging includes the connecting address and user, if available, as well as the request qname, which is unique for each operation.
<b>Web – UI/Runtime</b>	
PL-19646	Fixed an issue preventing the Guidewire application shortcut keys from working as intended. Previously, the keyboard responded only to the shortcut keys of the browser. Now the keyboard responds to the Guidewire application shortcut keys.
<b>Work Queues</b>	
PL-18675	Fixed several issues that involved the <code>Work Queue Info</code> screen on the <code>Server Tools</code> tab.
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 ClaimCenter is to process orphans before new items on this worker.
PL-19059	The <code>Work Queue Info</code> screen on the <code>Server Tools</code> tab now reports the last time that ClaimCenter processed a work item, showing either the last Success time or the last Exception time. If the time was for an exception, the <code>Work Queue Info</code> screen also reports the number of consecutive work items that resulted in an exception. The screen shows two other dates as well. The Last Notification time is the last time the worker woke up. If it found work, ClaimCenter also updates the Last Batch Found time.

## Known Issues and Limitations

This section describes known issues with this release of Guidewire ClaimCenter.

- “ClaimCenter Known Issues” on page 260

- “Studio/Platform Issues” on page 260

**Note:** For maintenance releases, Guidewire often defers 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.

## ClaimCenter Known Issues

### Implementing Policy Refresh

Before attempting to implement policy refresh, Guidewire strongly recommends that you contact Customer Support for guidance on implementation.

### New Claim wizard not saving selected vehicle in some cases (CLM-13973)

**Issue** – The second step of the New Claim wizard is not saving your vehicle selection when you leave the screen and then come back. For example, you choose a vehicle and then click the **Policy Details** link, and when you click **Basic Info** on the left to return to this screen, the vehicle is no longer selected.

**Workaround** – Reselect the vehicle when you return to the wizard screen.

### Upgrade from ClaimCenter 6 to ClaimCenter 7 does not work for SQLServer if MigrateToLargeIDsAndDatetime2 is set to true (CLM-17739)

**Issue** – During upgrade with SQLServer as the database, the `cc_tmppagglimitrpt` table fails to be upgraded if it is not empty and `MigrateToLargeIDsAndDatetime2` is true.

**Workaround** – If there is data in the table `cc_tmppagglimitrpt`, truncate the table before beginning the upgrade by running the following SQL command:

```
DELETE FROM cc_tmppagglimitrpt
```

### ABContactPlugin now requires ContactManager 7.0.1 or later (CLM-17990)

**Issue** – The class `gw.plugin.addressbook.ab700.PluginRetrieveOperation`, which is used by `ABContactPlugin`, has a method that requires the `ABContactAPI` web service in ContactManager 7.0.1 or later. This method, `retrieveReplacementContact`, will cause errors if your ClaimCenter application continues to be integrated with ContactManager 7.0.0. Guidewire recommends that you upgrade to ContactManager 7.0.1 or later. If you cannot upgrade ContactManager, see the workaround for this issue.

**Workaround** – Copy `ClaimCenter/modules/cc/gsrc/gw/plugin/addressbook/ab700/PluginRetrieveOperation.gs` to `ClaimCenter/modules/configuration/gsrc/gw/plugin/addressbook/ab700/PluginRetrieveOperation.gs`. Edit the copied file and remove the annotation `@ReadOnly`, the `retrieveReplacementContact` method, and all calls to the `retrieveReplacementContact` method. Then save the file.

## Studio/Platform 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, Internet Explorer 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 an Internet Explorer 9 workaround that reduces these issues. Change the Internet Explorer 9 **Accelerated Graphics** settings on the **Advanced** tab of the **Internet Options** dialog.

## 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 the `config.xml` file. 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:**

```
ContactManager/modules/pl/config/registry/compatibility-xsd.xml
```

To:

```
ContactManager/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 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** – Entity localization table names have a shorter, 16-character limit than other entity tables. If the localization table name exceeds the 16-character limit, the error message indicates incorrectly that 18 characters are allowed.

**Workaround** – Ensure that the localization `tableName` property specified in the entity extension file is less than 16 characters.

### 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.

### Problem with `regen-java-api` command and JAR files (PL-16351)

**Issue** – If you run the `ClaimCenter/bin/gwpc regen-java-api` command, ClaimCenter creates a `ClaimCenter/java-api/lib` directory and puts JAR files for Java development in this directory. However, ClaimCenter 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:

ClaimCenter/admin/lib

Copy them into the following directory:

ClaimCenter/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, ClaimCenter throws `ParseResultsException`. This is the intended behavior.

**Workaround** – Restart the workflow engine. To do so:

1. Log into ClaimCenter using an administrative account.
2. Access Internal Tools → Reload.
3. Click Reload Workflow Engine.



---

## chapter 29

# Guidewire ClaimCenter 7.0.1 Release Notes

## Release 7.0.1

**Document Revision:** 23-June-2016

Copyright © 2001-2016 Guidewire Software, Inc.

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, Guidewire Spotlight, Gosu, Adapt and succeed, 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:

- “Release Information” on page 266
- “Installing This Release” on page 266
- “Support” on page 266

- “Issues and Major Changes” on page 266
- “Improvements and General Issues” on page 274
- “Known Issues and Limitations” on page 286

## Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

### Version Number

This release of Guidewire ClaimCenter is 7.0.1.

## Installing This Release

Refer to the following:

- The *ClaimCenter Installation Guide* for general installation information
- Prior ClaimCenter 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 might affect your installation.

- “ContactManager Integration” on page 267
- “Geocoding Using Bing and MapPoint (PL-16708)” on page 267
- “Changes to Archive Functionality” on page 267
- “Catastrophe Heat Map Configuration” on page 268
- “Guidewire Standard Reporting” on page 268
- “Run ContactAutoSync and Clear Failed Work Items (CLM-17855)” on page 270
- “ClaimCenter unable to delete a contact from Address Book (CLM-17300)” on page 270
- “Integrating ClaimCenter 7.0.1 with PolicyCenter 7.0.0 or 7.0.1 (CLM-17669)” on page 270
- “Speed up claim search by denormalizing Contact fields (CLM-17885)” on page 271
- “Audit of Gosu classes (CLM-17516)” on page 271
- “Upgrade iso.xsd to version 5.3 (CLM-17681)” on page 272
- “ContactAutoSyncWorkItem update (CLM-17990)” on page 272
- “Base PCF File Changes” on page 273
- “Rules Changes” on page 273
- “Changes in This Release Provided in Upgrade Diff Report” on page 273

## ContactManager Integration

This release of ClaimCenter provides an integration with Guidewire ContactManager. For more information, see “Integrating ContactManager with Guidewire Core Applications” in the *Contact Management Guide*. For general information on ContactManager, see the *Guidewire Contact Management Guide*.

## Geocoding Using Bing and MapPoint (PL-16708)

Guidewire has added an implementation of the GeocodePlugin that connects to the Microsoft Bing Maps Geocode Service. The Bing Maps plugin implementation replaces the Microsoft MapPoint implementation.

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.

## Changes to Archive Functionality

**Note:** To increase performance, most customers find increased hardware more cost effective than archiving unless their volume exceeds one million claims or more. Guidewire *strongly recommends* that you contact Customer Support before attempting to implement archiving.

ClaimCenter now supports archiving a claim as a serialized stream of data. You can choose to store the data in a file, in a document storage system, or in a database (as a single large binary object). The basic format is an XML document.

**Note:** For more information, see “Archiving Claims” in the *ClaimCenter Configuration Guide*.

As part of this change, Guidewire has modified the criteria that determine whether a claim is eligible for archive. Rather than basing eligibility for claim archiving on how many days have elapsed since `Claim.CloseDate`, ClaimCenter now bases archiving eligibility on the new `Claim.DateEligibleForArchive` field. Specifically, for a claim to be archiveable, its `DateEligibleForArchive` field must be a non-null date and time that is not later than the current system date and time. Guidewire has also modified the list of configuration parameters:

- Added `config.xml` configuration parameter `DaysRetrievedBeforeArchive`.
- Removed `config.xml` configuration parameter `ArchiveServer`.

Guidewire has removed the following columns from all base configuration entities on which they existed:

- `ArchiveID`
- `Overlap`
- `PartitionID`
- `ArchiveAdmin`
- `ArchiveTypeKey`

As part of the upgrade to ClaimCenter 7.0.1, Guidewire removes the `AdminTable` delegate from customer extensions.

Guidewire has set `extendable="false"` and `final="true"` in the following entities in the base configuration:

- `ArchiveWorkItem`
- `ArchiveFailureDetails`

Guidewire has deleted the following entities from the base configuration:

- `ArchiveTypeKey`
- `ArchiveTransitionRecord`
- `ArchiveAdminKey`

Guidewire has added the following new `config.xml` configuration parameters for use with domain graphs and archiving:

- `DomainGraphKnownUnreachableTables` – Use to provide a comma-separated list of links with known issues. Format the list as `relativeName.linkAttributeName`.
- `DomainGraphKnownLinksWithIssues` – Use to provide a comma-separated list of known unreachable tables.

These configuration parameters specify a list of known cases in which a subgraph of the domain graph can potentially become unreachable if a foreign key is null. Adding a known issue or a table to the list removes it from the list of warnings on the **Warning** tab of the **Domain Graph Info** screen, which is on the **Server Tools** tab.

**Note:** For additional archive-related changes, see the **Archiving** section under the “Improvements and General Issues” on page 274.

## Catastrophe Heat Map Configuration

For details on how to configure the catastrophe heat map functionality, consult the Knowledge Base article *Configuring the ClaimCenter Catastrophe Dashboard*.

### See also

- “Catastrophe Policy Location Download” in the *Integration Guide*
- “Catastrophe Dashboard” in the *Application Guide*

## Guidewire Standard Reporting

---

**IMPORTANT** Perform the steps outlined in this release note before you start the installation and integration procedures listed in the *ClaimCenter Reporting Guide*.

---

In addition to the steps outlined in “Installing Guidewire Standard Reporting” (available in the *ClaimCenter 7.0.1 Reporting Guide*), perform the following steps before you perform the procedures listed in that guide:

- “Step 1: Update and Modify Class CognosPlugin.gs” on page 268
- “Step 2: Update and Modify CognosReportPage.pcf” on page 269
- “Step 4: Update and Modify Plugin CognosPlugin” on page 269
- “Step 4: Restart the Application Server” on page 269

### Step 1: Update and Modify Class CognosPlugin.gs

You must modify Gosu file `CognosPlugin.gs`.

1. Copy the following file:

`...\\modules\\pl\\gsrc\\gw\\plugin\\cognos\\CognosPlugin.gs`

to the following directory:

`...\\modules\\configuration\\gsrc\\gw\\plugin\\cognos`

---

**IMPORTANT** Do not modify this file in its original location. If you do so, you invalidate your ClaimCenter installation.

---

2. Verify that the file is writable. If not, make it writable.

3. Add the `@Exportable` annotation directly above the `CognosPlugin` class definition, for example:

```
@Exportable  
@Availability(AvailabilityLevel.MULTIUSER)  
@Distributed  
class CognosPlugin implements IStartablePlugin, InitializablePlugin {  
    ...
```

4. Add the following variable declarations in the `CognosPlugin` class

```
private static var _gatewayEndPointUrl : String  
private static var _dispatcherEndPointUrl : String
```

**5. Do the following:**

- a. Remove the `getHostNameAndPort` method and replace it with the following methods:

```
public static function getGatewayEndPointUrl() : String {
    return _gatewayEndPointUrl
}

public static function getDispatcherEndPointUrl() : String {
    return _dispatcherEndPointUrl
}
```

- b. Add the following statements in the `start` method's second `if` statement, which starts with the code  
`if (_state != StartablePluginState.Started):`

```
_gatewayEndPointUrl = _params.get("gatewayEndPointUrl") as String
_dispatcherEndPointUrl = _params.get("dispatcherEndPointUrl") as String
```

**Step 2: Update and Modify CognosReportPage.pcf**

You must modify the PCF file `CognosReportPage.pcf` as follows:

1. Open ClaimCenter Studio.
2. Find and open `CognosReportPage.pcf` for editing.
3. Select the `TemplatePanel` widget in the PCF page. This action opens the `Properties` pane at the bottom of the screen.
4. Copy the following `<iframe>` code and replace the existing text string in the `value` field:

```
<iframe id="cognosFrame" src="<%printContent(gw.plugin.cognos.CognosPlugin.getSetCognosCookiesUrl(), false)%>?gwGatewayEndPointUrl=<%printContent(gw.plugin.cognos.CognosPlugin.getGatewayEndPointUrl(), false)%>&gwDispatcherEndPointUrl=<%printContent(gw.plugin.cognos.CognosPlugin.getDispatcherEndPointUrl(), false)%>&gwNamespace=<%printContent(com.guidewire.pl.system.dependency.PLDependencies.getPluginConfig().getPluginDef("CognosPlugin").getParams().get("namespace"), false)%>&gwUser=<%printContent(com.guidewire.pl.system.dependency.PLDependencies.getCommonDependencies().getCurrentUser().getCredential().getUserName().toLowerCase(), false)%>&gwPassword=<%printContent(java.net.URLEncoder.encode(com.guidewire.pl.system.dependency.PLDependencies.getCommonDependencies().getCurrentUser().getCredential().getPassword()), false)%>" style="width:1400px;height:1050px"></iframe>
```

5. Restart Studio.

**Step 4: Update and Modify Plugin CognosPlugin**

You must add additional plugin parameters to `CognosPlugin`.

1. Open the following plugin registry:

`Plugins → gw → api → startable → IStartablePlugin → CognosPlugin`

2. Add the following parameters to the plugin:

Name	Example value
<code>dispatcherEndPointUrl</code>	<code>http://hostname1:9300/p2pd/servlet/dispatch</code>
<code>gatewayEndPointUrl</code>	<code>http://hostname1:80/ibmcognos/cgi-bin/cognos.cgi</code>

**Note:** You must set additional plugin parameters to complete the plugin configuration. See the *ClaimCenter 7.0.1 Reporting Guide* for details. See especially “Step 4: Configure the Cognos Plugin” under “Installing Guidewire Standard Reporting”.

**Step 4: Restart the Application Server**

Restart the application server. It is also useful to restart Studio before you start the application server to discover any errors with copy and paste or any typing errors.

After you restart the application server, continue with the installation instructions outlined in the *ClaimCenter Reporting Guide*.

## Run ContactAutoSync and Clear Failed Work Items (CLM-17855)

The ContactAutoSyncWorkItem has been modified to use a soft reference to the Contact table instead of a foreign key. Before upgrading, the ContactAutoSyncWorkItem table must be empty. You must run the ContactAutoSync worker and clear its ContactAutoSyncWorkItem table of any failed items.

### To run the ContactAutoSync work queue and clear the work item table

1. Start ContactManager and ClaimCenter.
2. Log into ClaimCenter.
3. In ClaimCenter, press Alt+Shift+T. ClaimCenter opens the **Server Tools** tab.
4. Under **Actions** on the left, click **Work Queue Info**.
5. For the **ContactAutoSync** worker, click the **Notify Workers** button to run the work queue.
6. Under **Actions** on the left, click **Batch Process Info**.
7. For the **Purge Failed Work Items** batch process, click the **Run** button to clear failed items from all work queues.
8. The previous steps might not successfully delete all rows from the cc\_autosyncworkitem table if work items failed very recently. In this case, stop your server and run the following SQL to delete the rows directly:

```
DELETE FROM cc_autosyncworkitem WHERE status=3
```

## ClaimCenter unable to delete a contact from Address Book (CLM-17300)

ClaimCenter now attempts to retire contacts in the Contact table when references to them are modified. When any foreign key links to the contact to be removed are changed, a work item is created for the new Contact Retire work queue to check later to see if the contact can be retired. Additionally, if ContactManager calls the ClaimCenter web service `ContactAPI.isContactDeletable`, ClaimCenter can attempt to delete the contact.

You might need to configure this feature if you have implemented data model extensions that have foreign keys to contacts. You can use the plugin implementation `gw.plugin.contact.RetireContactPlugin` to add properties to a list of safe properties, properties that do not prevent a contact from being deleted. Additionally, your entity extensions can implement `ContactRetireBean` if they have an unsafe foreign key link to a contact. The implementation can determine if an entity instance will prevent a contact instance from being retired and, if not, if any other beans can be retired along with the contact. See the class `gw.api.contact.ContactRetireBeanImpl` for an example of checking to see if a `ContactContact` and the associated `Contact` can be retired.

For more information, see:

- The Gosu API Reference in the Studio Help menu
- “Contact Retire” in the *System Administration Guide*

## Integrating ClaimCenter 7.0.1 with PolicyCenter 7.0.0 or 7.0.1 (CLM-17669)

### To integrate ClaimCenter 7.0.1 with PolicyCenter 7.0.0 or PolicyCenter 7.0.1

1. Open Studio and click **RPC-Encoded Web Service** in the **Resources** pane.
2. Click `pcintegrationPC702`.
3. Make sure that your version of PolicyCenter, 7.0.0 or 7.0.1, is running.
4. Click the Refresh button to the right of the URL to refresh the WSDL for this web service.

This step enables the integration named pcintegrationPC702 to use WSDL from your running version of PolicyCenter 700 or 701 instead of PolicyCenter 7.0.2.

5. Edit `gw.plugin.pcintegration.pc702.PolicySearchConverter` in Studio and delete the following lines of code:

```
pcCriteria.PrimaryInsuredCity = ccCriteria.InsuredAddress.City  
pcCriteria.PrimaryInsuredState = ccCriteria.InsuredAddress.State.Code  
pcCriteria.PrimaryInsuredPostalCode = ccCriteria.InsuredAddress.PostalCode  
pcCriteria.PrimaryInsuredCountry = ccCriteria.InsuredAddress.Country.Code
```

6. Replace the deleted code with:

```
pcCriteria.State = ccCriteria.InsuredAddress.State.Code
```

7. Save your changes and restart ClaimCenter.

## Speed up claim search by denormalizing Contact fields (CLM-17885)

This change adds new fields that are denormalizations of existing fields. ClaimCenter automatically maintains the fields during normal system operation or when loading from staging tables. The benefit of the new fields is that searching for claims by name on both the simple and advanced claim search screens now scales better on large databases. These searches will run faster and with less I/O on the database server.

You will probably notice the following changes:

Upgrading to 7.0.1 will take more time because the new columns have to be populated. In Guidewire testing, the extra cost was around 40 to 50 minutes for a database containing 2.5 million claims, a fairly large database. This extra cost might be noticeable if you are upgrading from 7.0.0. However your database will probably be smaller than the test database, so the time will be shorter. If you upgrade from an earlier release, you might have a larger database and see a bigger effect. But for earlier releases, database upgrade time is likely to already be fairly substantial, so this extra step is not likely to make a big difference.

The `cc_claimcontact` table in the database now has extra columns and extra indexes on those columns. If you have added your own indexes to `cc_claimcontact`, check to see if the indexes need to be altered as a result of the new base configuration.

There is a new batch process, `populatesearchcolumns`, that you can use to populate the new denormalized columns if they get out of sync with their source columns. It is not likely that you will have to use this batch process, but it is available for emergencies. The batch process is not visible on the Batch Process Info screen, and in the base configuration, you cannot schedule the process, although you can configure it differently.

You can run the batch process on the command line in the `ClaimCenter/admin/bin` directory by using the command `maintenance_tools -startprocess populatesearchcolumns`. Guidewire recommends that you run it during a maintenance period. The batch process can take a substantial amount of time on a large database because it updates a lot of rows. In Guidewire's testing, the process took as much as 90 minutes on Oracle and SQLServer databases with 2.5 million claims.

## Audit of Gosu classes (CLM-17516)

Gosu classes and enhancements that were not previously marked with the `@Export` or `@ReadOnly` annotation have now been marked appropriately. For classes, the default visibility is `hidden`, so adding `@ReadOnly` or `@Export` to a class that was not previously annotated simply increases visibility and has no impact on your existing class extensions.

For enhancements, the default annotation was `@Export`. Therefore, any enhancements that are now marked `@ReadOnly` could have an impact on upgrade. If you made changes to those enhancements before upgrade, Guidewire recommends that you revert those files to base and put your changes in other enhancements.

The following list shows enhancements in ClaimCenter 6.0.6 that were marked @ReadOnly in 7.0.1:

```
NameTranslatorImplEnhancement.gsx  
DocumentCreationInfoEnhancement.gsx  
GWBenefitsEnhancement.gsx  
GWClaimIndicatorEnhancement.gsx  
GWLARGELOSSThresholdEnhancement.gsx  
GWVehicleEnhancement.gsx  
GWBaseBundleEnhancement.gsx  
GWDateEnhancement.gsx
```

## Upgrade iso.xsd to version 5.3 (CLM-17681)

ClaimCenter 7.0.1 uses ISO v5.3 and has the following iso.xsd changes.

The following are new elements that can occur under the com.iso\_CovInfo1 element in com.iso\_AddCovInfo:

```
com.iso_DeleteFromCMS (IXMLNode)  
com.iso_NotSendCovCMS (IXMLNode)
```

The following new element can occur under ClaimInvestigationAddRq:

```
com.iso_RecallRqInd (IXMLNode)
```

There is a new IXMLNode, com.iso\_DeleteFromCMS, under the com.iso\_CovInfo1 element under com.iso\_AddCovInfo under ClaimInvestigationAddRq.

There is a new IXMLNode, com.iso\_StopCMSQuery, under com.iso\_SIUParty.

The ClaimsOccurrence element under ClaimInvestigationAddRq now has an array of ProbableIncurredAmt elements called ProbableIncurredAmts. A ProbableIncurredAmt element contains an optional currency. It was previously limited to a single occurrence.

SuitFiledInds and ClosedDts are now arrays on ClaimsPartyInfo. They were previously limited to a single occurrence.

New elements under InvestigationInfo:

- com.iso\_TheftTypeInd is a new String type under InvestigationInfo.
- com.iso\_VehRecalls is a new array of com.iso\_VehRecall instances (of type IXMLNode) under InvestigationInfo.
- com.iso\_SFNMVTISs is a new array of com.iso\_SFNMVTIS instances (of type IXMLNode) under InvestigationInfo.

## ContactAutoSyncWorkItem update (CLM-17990)

This change enables ContactAutoSyncWorkItem to get the latest AddressBookUID for a merged contact. It involves a call to a method available only in ContactManager 7.0.1 and later. If you upgrade to ContactManager 7.0.1, you do not need to do anything.

If you are integrating ClaimCenter 7.0.1 or later with ContactManager 7.0.0, you need to make the following changes to the file PluginRetrieveOperation.gs.

1. Copy ClaimCenter/modules/cc/gsrc/gw/plugin/addressbook/ab700/PluginRetrieveOperation.gs to ClaimCenter/modules/configuration/gsrc/gw/plugin/addressbook/ab700/PluginRetrieveOperation.gs.
2. Edit the copied file and remove the annotation @ReadOnly, the method retrieveReplacementContact, and all calls to the retrieveReplacementContact method.
3. Save the file.

## Base PCF File Changes

All links below require the `ReleaseNotes_files` directory on your local disk.

### **ClaimCenter release 7.0.0 to 7.0.1**

- To view a report of the changes in the base PCF files in the `modules/cc` directory, [click here](#).
- To view a report of the changes in the base PCF files in the `modules/p1` directory, [click here](#).

## Rules Changes

### **ClaimCenter release 7.0.0 to 7.0.1**

- To view a report of the changes in the base rules in the `modules/cc` 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

The following are the primary improvements and issues corrected in this release:

ID	Description
<b>Activities and Workplan</b>	
CLM-14863	Fixed a problem that occurred if an activity showed up on a user's desktop activity page, but they did not have permission to view it. In earlier versions of ClaimCenter, clicking the activity would cause a stack trace error in the user interface. In ClaimCenter 7.0, a fix was introduced that would hide the link to such activities, but this fix proved to be a performance problem. So, in ClaimCenter 7.0.1 the fix was amended to show the link, but if the user clicks it, they see the error message "You do not have permission to view this Activity". This fix, while not as simple for the user, provides a big performance gain on this heavily used page because ClaimCenter does not have to check the claim access control list for every claim mentioned on the activity desktop page. Also, it rarely happens that a user's desktop has activities that the user does not have permission to view.
CLM-15738	When you assign an activity through the activity detail worksheet, the assignment popup now appears in the workspace. The activity worksheet also now remains open after you assign the activity. This fix corrects problems with starting to assign an activity but not completing the assignment, and subsequently being unable to access the activity from the Claim Summary screen.
<b>Administration, Catastrophe, Claim Associations, Dashboard, Desktop, Team, Question Sets</b>	
CLM-17362	The <code>EmailAddress</code> field on <code>AutomatedNotificationHandler</code> has been made nullable to enable extensions of <code>SpecialHandling</code> that do not set the email address when the handler is created. For example, the email address could be for a contact filling a particular role on a claim, which would be added later. As part of this change, a large section of Gosu code called to save an email as a Document on the <code>Claim</code> , which was common between <code>CreateEmailScreen.pcf</code> and <code>AutomatedNotificationHandlerMethodsImpl.gs</code> , was moved to an enhancement on <code>Claim</code> , <code>GWCclaimEmailEnhancement</code> .
CLM-17801	The Team tab has <code>As of</code> fields showing when team statistics were last calculated. These fields now show the correct date and time on all servers within the cluster.
<b>Application Integration</b>	
PL-17945	Guidewire has changed the type of the URL field in the <code>suite-config</code> XSD definition file from <code>xs:anyURI</code> to <code>xs:string</code> .
<b>Archiving</b>	
PL-18992	As part of the changes made for file-based archiving, Guidewire now requires that you increment the version number stored in <code>extensions.properties</code> 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 <code>suite-config.xml</code> . Guidewire defines duplicate entries as those with the same name, URL, and env values.
CLM-13149	Removed a number of fields from the <code>ClaimInfoSearchView</code> view entity because ClaimCenter uses this view entity only to display information about archived claims. For those claims, the following removed fields are almost always null: <code>Flagged</code> , <code>ActiveAdjuster</code> , <code>ActiveInsured</code> , <code>ClaimCurrency</code> , <code>FuturePayments</code> , <code>RemainingReserves</code> , <code>TotalPayments</code> , <code>ClaimNumberOrder</code> , <code>ClaimStateOrder</code> , <code>LossDateOrder</code> , and <code>PolicyNumberOrder</code> .  Also modified the PCF files <code>ClaimSearchResultsLV.pcf</code> and <code>ArchiveClaimRetrievePopup.pcf</code> to no longer display these fields. All display keys that were used to reference these fields in the user interface have also been removed.
CLM-16553	Fixed an issue where the popup you use to search for claims when editing a claim association was sometimes searching for archived claims. This fix ensures that the claim association screen's claim search popup searches only for active claims.
CLM-16848	Changing the policy or risk units in the New Claim wizard leads to residual data links that can prevent claims from being archived. This issue also occurs when refreshing a policy in ClaimCenter 6.0.x, and it can affect claims upgraded to 7.0. Running the new batch process <code>Retired Policy Graph Disconnector</code> removes those links and enables the claims to be archived.

CLM-17228	Fixed an issue in which ClaimCenter was archiving Addressbook fingerprints. This could lead to contacts being erroneously out of synchronization if the claim was restored.
CLM-17229	Fixed an issue that caused an <code>IllegalArgumentException</code> if you attempted to retrieve an archived claim multiple times.
CLM-17235	Modified the criteria that determine whether a claim is eligible for archiving. Rather than basing eligibility for claim archiving on how many days have elapsed since <code>Claim.CloseDate</code> , ClaimCenter now bases archiving eligibility on the new <code>Claim.DateEligibleForArchive</code> field. Specifically, for a claim to be archiveable, its <code>DateEligibleForArchive</code> field must be a non-null date and time that is not later than the current system date and time. For more details, see "Selecting Claims for Archive Eligibility" in the ClaimCenter Configuration Guide.  This fix added two new archive rules:  <b>ARC05000 - Incomplete Review Rule</b> – Instructs the archive process to ignore claims with incomplete reviews. <b>ARC0a</b> <b>6000 - Unsynced Review Rule</b> – Instructs the archive process to ignore claims with vendor reviews that have not already been synchronized with the Address Book.
CLM-17237	Added a method, <code>ClaimInfoUtil.newArchivedClaimInfo</code> , to support creation of <code>ClaimInfo</code> objects that correspond to claims that have already been archived. The new method creates a new instance of <code>ClaimInfo</code> , sets its <code>ArchiveState</code> to <code>archived</code> , and sets the non-scriptable fields on the entity based on the arguments.
CLM-17242	Added Archived and Restored filter options to the Claim History drop-down list.
CLM-17245	The base ClaimCenter configuration now requires that a user enter a comment before restoring a claim.
CLM-17249	Added a subject for Notes created by archiving issues.
CLM-17337	Removed the <code>MessageHistory</code> entity from the claim graph. This change affects ClaimCenter 7.0.1 and greater. A claim archived in a version of ClaimCenter before 7.0.1 can still include <code>MessageHistory</code> entities, but when such a claim is retrieved in 7.0.1 or later, those entities will be deleted.  Even though they are no longer part of the graph, <code>MessageHistory</code> entities will still be deleted when the linked claim, if any, is marked for purge.
CLM-17610	There was a bug that caused ClaimCenter to throw an exception when a user who was not assigned to any group tried to restore a claim that they had created. Now if a user who is not in any group tries to restore a claim, ClaimCenter assigns an activity for restoring the claim to a user and group.

**Archiving Claim - FNOL Wizard, New Claim Wizard, Claim Snapshot, FNOL mapper**

CLM-17692	Changing the policy or risk units in the New Claim wizard leads to residual data links that can prevent claims from being archived. This issue also occurs in 6.0.x Policy Refresh, and can affect claims upgraded to 7.0. The <code>RetiredPolicyGraphDisconnector</code> batch process removes those links.
-----------	---

**Archiving Financials - Aggregate Limits, Policy Period**

CLM-17776	Purging of claims with aggregate limits was not being handled appropriately. ClaimCenter now ensures that a claim or archived claim that has aggregate limits cannot be purged. If you try to purge a claim with aggregate limits, you see the error message, "Cannot purge a claim with aggregate limits."
-----------	---

**Archiving Financials Misc**

CLM-17254	Added a new column on the Bulk Invoice Item list view that indicates whether the claim associated with the invoice item has been archived. ClaimCenter hides this column if none of the associated claims are archived.
-----------	---

**Assignment**

PL-13175	Delegated the <code>UserRoleAssignment</code> entity out to a <code>UserRoleAssignmentDelegate</code> and removed the <code>UserRoleAssignment</code> entity at the platform level. All Guidewire applications now own their own <code>UserRoleAssignment</code> entity and use entity delegation to <code>UserRoleAssignmentDelegate</code> .
PL-18196	Fixed <code>assignGroupByRoundRobin</code> to manage <code>GroupAssignmentState</code> with proper set of keys including intended <code>GroupType</code> to assign a group properly.
PL-18397	Fixed an issue that occurred during the creation of an exposure at the end of the New Claim wizard if you were using <code>assignGroupByRoundRobin</code> assignment.

**Assignment, Email, Notes, Documents**

CLM-14742	Changed the exception handling for the calls to <code>IDocumentMetadataSource.saveDocument</code> and <code>IDocumentMetadataSource.removeDocument</code> to display exception messages in the user interface rather than only in the logs. If there is a problem saving a document, it is now possible to let the user know.
-----------	---

**Batch Processes**

PL-13912	Guidewire has added the ability to set an env attribute on the <code>&lt;ProcessSchedule&gt;</code> element in <code>scheduler-config.xml</code> 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 a Last Run Status column to the Processes table on the <b>Batch Process Info</b> screen. This column enables 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"> <li>• Completed – The last run has completed successfully.</li> <li>• Not available – This task has not ran yet or running right now.</li> <li>• Failed/Interrupted –The last run has failed or was interrupted. See the history of the process for more details.</li> </ul>

**Claim - FNOL Wizard, New Claim Wizard, Claim Snapshot, FNOL Mapper**

CLM-14078	In the New Claim wizard, when you choose a policy and click <b>Next</b> in the first step, you might receive a “Duplicate Claim” warning. Previously, you had to click <b>Close</b> on the warning and then click <b>Next</b> again to proceed with claim setup. The problem with having to click <b>Next</b> again was that it caused ClaimCenter to retrieve the policy again, which was unnecessary, since it had already retrieved the policy. Now, the duplicate claim warning still appears in the workspace, but when you click <b>Close</b> in the warning message, you can simply proceed with the wizard.
CLM-15400	If you choose a policy in the New Claim wizard and add data such the loss date and continue past Step 1: <b>Search or Create Policy</b> , and then while still in the wizard you go back to Step 1 and change the loss date, ClaimCenter now resets the New Claim wizard and resets the policy on the new claim. This behavior ensures that you have current policy data in the claim.
CLM-17392	In the base configuration, if you added a pedestrian in the New Claim wizard, it was previously possible to choose a Company subtype, which caused errors. Your choice of contacts is now correctly limited to Person subtypes. You can configure this setting in <code>entityroleconstraints-config.xml</code> .
CLM-17457	It is now possible to configure a RiskUnit with a DisplayName that has a nonnegativecurrencyamount column, and have it display without throwing an exception.
CLM-17547	Changing the policy or risk units in the New Claim wizard leads to residual data links that can prevent claims from being archived. This issue also occurs in 6.0.x Policy Refresh, and can affect claims upgraded to 7.0. The <code>RetiredPolicyGraphDisconnectorBatch</code> batch process removes those links.
CLM-17594	<p>To prevent <code>ConcurrentDataChangeException</code> exceptions, if the user is working on a claim, such as in the Parties Involved screen, ClaimCenter now prevents the user from editing a contact on the <b>View in Address Book</b> screen. Instead, the user must edit the contact in the context of the claim and then can use the <b>Save to Claim and Address Book</b> button to update the contact in the Address Book. The list of PCF files changed is in the bug.</p> <p>Also there is a new method, <code>NewClaimWizardUtil.isInNewClaimWizardDraftSave</code>, that determines if the current bundle commit is part of a save for an intermediate draft claim in the New Claim wizard. You can use this method to control behavior in rules or other code to prevent actions that might cause a <code>ConcurrentDataChangeException</code>.</p>
CLM-17457	It is now possible to configure a RiskUnit with a DisplayName that has a nonnegativecurrencyamount column, and have it display without throwing an exception.

**Claim Metrics**

CLM-17619	<p>Changed Claim Health Metrics to truncate the calculated percentage value of any <code>PercentMetric</code> that exceeds the size of the percent field, as defined by <code>PercentageMetricDataType</code> in <code>datatype.xml</code>.</p> <p>Also changed Claim Health Metrics to reduce the precision of any <code>PercentMetric</code> that exceeds the configured amount.</p>
-----------	--

**Claim - Summary, Loss Details, Exposures, Incidents, Litigation, Etc.**

CLM-13064	In the New Claim wizard, if <code>Wizard.saveDraft</code> is called when there are empty required fields, validation errors will be shown. Previously, trying to save a draft of a claim with empty required fields caused an exception and showed a stack trace rather than showing validation errors.
-----------	---

CLM-16019	The SIUScore field on the Claim Status page—the Special Investigation Unit Score—now shows the correct value. It now uses <code>util.SIUTotalScore.getSIUTotalScore(indicator.Claim)</code> to get the value rather than <code>indicator.claim.SIScore</code> .
<b>Clustering</b>	
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, ClaimCenter 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.
<b>Code Utilities</b>	
PL-18054	If you have used the \${x} substitution in document templates, then you need to revert those uses back to the <%= ... %> syntax.
<b>Cognos Integration</b>	
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 ClaimCenter database. (Guidewire normalizes user names to all lower case in its communications with Cognos.)  There is still an issue with Cognos not being able to distinguish between Guidewire users that differ only in case even if these users are distinguished in Guidewire. Do not create Guidewire LDAP users that differ only in case.
<b>Command-line Tools, Toolkit, Data Dictionary, External Entities, CC Batch Process</b>	
CLM-13617	Additivity statements in <code>logging.properties</code> were corrected. An additivity statement attempts to keep the category from sending its log messages up to its parent, therefore preventing duplicate log messages across files.
CLM-14188	Removed the MIME type descriptions from <code>config.xml</code> in the base configuration, since the preferred place to configure them is in the <code>Mimetype</code> display keys. For example, to set the description for the <code>video/mpeg</code> MIME type, edit the value of the <code>Mimetype.video_mpeg</code> display key. Also, the description for the <code>image/pjpeg</code> type has been edited to differentiate it from <code>image/jpeg</code> .
CLM-17260	Fixed an issue in which ClaimCenter displayed an exception if an invalid claim number was specified for <code>-scheduleforarchive</code> .
<b>Configuration Upgrade</b>	
PL-18021	Fixed an issue with how the Upgrade tool automatically handled rule upgrades.
<b>Contacts, Roles</b>	
CLM-17863	The <code>InstantaneousContactAutoSync</code> setting in <code>config.xml</code> is now <code>false</code> in the base configuration, causing contacts to be synchronized only when the <code>ContactAutoSync</code> batch process runs twice daily at 5:00 am and 5:00 pm. If you set <code>InstantaneousContactAutoSync</code> to <code>true</code> , the <code>ContactAutoSync</code> worker activates automatically when autosync events are received. This change was made because instantaneous synchronizations would sometimes cause <code>ConcurrentDataChangeException</code> (CDCE) exceptions.
CLM-17864	The <code>ContactAutoSync</code> batch process needs to run before the <code>Financials Escalation</code> batch process to prevent the financials from hitting validation errors because the contacts are out of sync. In the base configuration, the <code>Financials Escalation</code> process runs daily at 6:05 am and 6:05 pm, so <code>ContactAutoSync</code> has been changed to run at 5:00 am and 5:00 pm daily.  This change was made because the <code>InstantaneousContactAutoSync</code> configuration parameter in <code>config.xml</code> is now <code>false</code> in the base configuration, causing contacts to be synchronized only when the <code>ContactAutoSync</code> batch process runs. If you set <code>InstantaneousContactAutoSync</code> to <code>true</code> , the <code>ContactAutoSync</code> worker activates automatically when autosync events are received. However, the worker might activate while a user is editing that contact, making it impossible for them to save their changes. That is why <code>InstantaneousContactAutoSync</code> is <code>false</code> by default.

CLM-17931	ClaimCenter now prevents you from simultaneously editing a contact in the Address Book and in another part of the application. For example, while editing a contact in the <b>Parties Involved</b> screen, if you click the <b>View In Address Book</b> button, when the contact opens in the Address Book, you cannot edit the contact there. On the <b>Parties Involved</b> screen, you can use the <b>Save to Claim and Address Book</b> button to save your edits to the Address Book.
<b>Contacts, Roles - Integration (AB, PC, BC, ISO, Metro), Policy, Reinsurance, Policy Refresh, LOB Configuration</b>	
CLM-17478	It is now possible to add electronic file transfer information to an insured contact and save the change to the claim and the Address Book. This action no longer causes a duplicate key error.
<b>Database</b>	
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.
<b>Database Upgrade</b>	
PL-17732	Fixed performance issues with the upgrade process.
PL-17849	Guidewire has added a <code>DateFromDatetime</code> method to the <code>BeforeUpgradeDBFunction</code> 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 <code>BeforeUpgrade</code> query, you can use this function in a <code>SELECT</code> statement, and it can take a nested function as an argument.
<b>Documentation</b>	
CLM-17219	It is possible that after you finish the New Claim wizard, any contacts brought over from a policy system or ContactManager might continue to appear to be out of sync for some time. When the wizard finishes, ClaimCenter sends messages to ContactManager to update any changed contacts. Until these messages complete and the update occurs in ContactManager, these contacts will appear to be out of sync.
<b>Entities/Metadata</b>	
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 <b>Entity Path</b> 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.
<b>Financials - Approvals, Authority Limits</b>	
CLM-17916	Fixed an issue in the Payment Exceeds Reserves (PER) authority limit for payments, where evaluation of a supervisor's authority limits double counted the amount of the payment.
<b>Financials - Bulk Invoices</b>	
CLM-17895	Fixed an issue in which ClaimCenter could potentially throw an exception if transferring a multicurrency check from one claim to another and the check had deductions. It is possible for this to happen if the original claim had a claim currency not equal to the reporting currency, and if the currency of the target claim was equal to the reporting currency.  Now, instead, ClaimCenter copies the <code>Deduction ReportingAmount</code> to the <code>ClaimAmount</code> . This is done because the claim and reporting currencies are equal, so these amounts must be equal. Guidewire also modified <code>CheckTransferDV.pcf</code> so that the user does not have to pick an exchange rate during this type of transfer. ClaimCenter always ignores the exchange rate in this type of transfer.
<b>Financials - Checks, Payments, Deductibles</b>	
CLM-16183	When adding additional payees using the <code>CheckCreator</code> , if you do not call the method <code>withPayTo</code> , the payee's <code>DisplayName</code> will now be used instead of the payee's <code>Name</code> .
CLM-17067	You can now clone a multi-payee manual check and change the recurrence from single to recurring. Previously, these actions caused a serious error in the clone check screen.
CLM-17648	Removed the restriction that manual checks cannot have a negative amount. Manual checks can now be negative.

CLM-17709	Fixed a bug with several methods of <code>gw.api.financials.CheckPayeeInfo</code> . Values supplied to the methods <code>withPayTo</code> , <code>withMailToAddress</code> , and <code>withRecipient</code> were being ignored for additional payees when using <code>gw.api.financials.CheckCreator</code> . These values are now saved and usable in <code>CheckCreator</code> . Also, if these methods are not called, whether for additional or primary payees, default values are now used for these payees in the created check. The default values supplied are the payee's <code>DisplayName</code> for <code>withPayTo</code> , the <code>DisplayName</code> of the payee's primary address for <code>withMailToAddress</code> , and the payee's <code>DisplayName</code> for <code>withRecipient</code> .
CLM-17752	Fixed a problem with the Reportable Amount being cleared out when going back and forth within the Check Wizard.
CLM-17826	Previously, if you tried to transfer a check that was a final payment on a claim to a claim or exposure that could not be closed (for example, due to open activities), ClaimCenter threw an exception. ClaimCenter now displays a message indicating why the target claim or exposure could not be closed.
CLM-17835	The <code>CheckCreator</code> object has an API that you must call in a certain order because you must create a check before you can prepare it for commit. If you call the methods out of order or multiple times, you now get an exception. The correct order is either to first call <code>createCheck</code> and then call <code>prepareForCommit</code> or to just call <code>createAndPrepareForCommit</code> . You can get a reference to a <code>CheckCreator</code> object in Gosu by calling either <code>claim.newCheckCreator</code> or <code>exposure.newCheckCreator</code> .
CLM-17992	Fixed an issue on the Payment Recode screen that caused ClaimCenter to display validation warnings and errors on the main page instead of in a worksheet, as is usually the case with validation messages.

**Financials - Checks, Payments, Deductibles, Reserves**

CLM-17896	Three new configuration parameters were added to <code>config.xml</code> in the Financial Parameters section. They enable you to configure the rounding modes used during calculation of financial transaction amounts. The new parameters are <code>DefaultRoundingMode</code> , <code>ReserveRoundingMode</code> , and <code>PaymentRounding</code> . For more details, refer to the documentation and the comments in <code>config.xml</code> above these new parameters.
-----------	--

**Financials - Checks, Payments, Deductibles, Reserves, ReserveLines, TAccounts, Financials Calculations**

CLM-18016	Fixed an error where split payment recode of a payment from sample data would throw an exception. Fixed the sample data to set up the exchange rates on transactions correctly based on the <code>MulticurrencyDisplayMode</code> setting in <code>config.xml</code> .
-----------	--

**Financials - DBChecks - Consistency/Integrity, Staging Tables, LoaderCallbacks**

CLM-13177	Added new integrity and consistency checks for the entities <code>CheckPortion</code> , <code>TransactionLineItem</code> , and <code>Deduction</code> . For each entity, added checks for the following cases: <ul style="list-style-type: none"> <li>• Reporting amount equals transaction amount if reporting currency equals transaction currency.</li> <li>• Transaction amount equals claim amount if transaction currency equals claim currency.</li> <li>• Claim amount equals reporting amount if claim currency equals reporting currency.</li> </ul>
-----------	--

**Financials - Reinsurance**

CLM-15124	Modified the <code>ReinsuranceDemoPlugin</code> and <code>RIDataGenerator</code> to support multiple Agreement Groups on the same claim, based on exposure coverage type. This change makes it easy to show multiple agreement groups in product demonstrations.
CLM-17168	Fixed an issue where operations involving regular financial transactions (such as setting reserves) would fail because automatically generated reinsurance transactions in the bundle violated some reinsurance constraints. Such operations are now allowed to succeed, and the rules that enforce the constraints now create an activity for a reinsurance manager to review the new reinsurance amounts.
CLM-17438	On the Reinsurance Summary screen, agreements are now sorted in the order of their applicability: first by subtype, then by attachment point. This sort order matches the order used in PolicyCenter.
CLM-17485	Added exception handling to the Reinsurance Exposures screen, allowing it to recover and restart properly when the Reinsurance Plugin fails.
CLM-17562	ClaimCenter has a new rule in the <code>RIAgreementValidationRules</code> rule set. This rule, <code>RIAV03000 - Proportional Agreement Percentage Total</code> , ensures that the total of all the proportional shares of the agreements in an agreement group cannot exceed 100% under any circumstances.
CLM-17845	Previously, when retrieving reinsurance information from PolicyCenter, if there was an error, ClaimCenter would throw an exception rather than displaying an error message. Now when PolicyCenter encounters an error with reinsurance information, ClaimCenter can display errors to the user, log errors, and create activities for the reinsurance manager to correct the problem.

CLM-17925	If you mark a claim with two or more agreement groups for reinsurance, there is now one note created to indicate that you performed this action. ClaimCenter no longer creates duplicate <b>Claim Marked for Reinsurance</b> notes in this case.
CLM-17935	If a reinsurance agreement was obtained from an external source, such as PolicyCenter, you are no longer allowed to edit it. You must modify the data in the external system and then manually retrieve reinsurance information again.
CLM-17942	The previous <b>Top Of Layer</b> field on the <b>Add Reinsurance Agreement</b> screen has been relabeled <b>Coverage Limit</b> in the user interface to match the same labeling in PolicyCenter.
CLM-17963	When ClaimCenter retrieves reinsurance coverage from PolicyCenter, PolicyCenter can report that the exposure no longer has coverage or throw an error indicating that ClaimCenter's coverage information might be out-of-date. In either case, ClaimCenter now removes the exposure from any reinsurance agreement group it belongs to.
CLM-17991	If a claim's policy is unverified or an exposure is not linked to a coverage entity on the policy, there is most likely incorrect or insufficient information to retrieve reinsurance information. ClaimCenter no longer retrieves reinsurance coverage in either of these cases.

**Financials - Reinsurance Integration (AB, PC, BC, ISO, Metro)**

CLM-17862	A bug was fixed in PolicyCenter 7.0.3 that prevented ClaimCenter from retrieving data from PolicyCenter about Per Risk and Annual Aggregate reinsurance agreements. To fix this bug in PolicyCenter 7.0.2, you can edit the PolicyCenter 7.0.2 gw.plugin.reinsurance.PCReinsurancePlugin class in Studio and replace its code with the code from the PolicyCenter 7.0.3 PCReinsurancePlugin class.
-----------	--

**Financials - ReserveLines, TAccounts, Financials Calculations**

CLM-17710	Methods on gw.api.financials.FinancialsCalculationUtil were deprecated prematurely. Methods that return a <b>FinancialsCalculation</b> object, such as <b>FinancialsCalculationUtil.getAvailableReserves</b> , are no longer marked as deprecated in ClaimCenter 7.0.1. You do not need to change code that uses these methods.
-----------	---

**Financials - SOAP APIs, Plugins, Rules**

CLM-17744	On the Recovery entity, the <b>acknowledgeVoid</b> , <b>acknowledgeRecode</b> , and <b>acknowledgeTransfer</b> methods now update the transaction status on the recovery's offsets and onsets to the appropriate acknowledged status. If the status of the offset or onset is <b>Submitting</b> , it is updated to <b>Submitted</b> . This change was made to match the behavior of these same methods on the Payment entity.
-----------	---

**Financials Misc**

CLM-13392	Fixed an issue where certain benefits fields for Personal Injury Protection (PIP) exposures would sometimes be displayed with a currency other than the claim currency.
CLM-14903	When you print a claim, if there is a manually entered exchange rate, it now prints out correctly.
CLM-16451	Added an upgrade trigger to correct an issue with <b>ReserveLineWrapper.ReserveLine</b> foreign keys that incorrectly refer to <b>ReserveLine</b> entities on another claim. The data issue corrected by this upgrade trigger can be caused by a bug in transferring checks associated with a bulk invoice, and its effect is to prevent a claim from being archived.
CLM-16613	Claims that have at least one transaction that is <i>not</i> in any of the following statuses are skipped during archive: submitted, rejected, voided, stopped, recoded, transferred, denied. For example, a claim with a payment that is awaiting submission will not be archived.
CLM-17899	A bug was limiting the loss types to which you could apply subrogation. Previously, you could apply subrogation only to auto loss types. You can now apply subrogation to all loss types, not just auto.
CLM-17962	Fixed a spelling error in the method <b>isDeleteable</b> . This old spelling of the method is now deprecated, and the new method is <b>isDeletable</b> . The method is on Transaction, BulkInvoice, Check, and CheckGroup.

**Geocoding/Proximity Search**

PL-17675	Fixed an issue in which the Bing Geocode plugin implementation was not visible in Studio.
----------	---

**Gosu**

PL-16935	Fixed an issue in which ClaimCenter threw a Null Pointer Exception if you attempted to change Gosu code in Studio and Studio 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: <i>'The property [or method], "Xxxx", is not visible under the parser's visibility constraints.'</i>
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.
<b>GX Tools</b>	
PL-12128	Added the ability for a web service to use types from a different WSDL. For example, <pre>&lt;xss:import namespace="http://example.com/gw/api/test/TheirAPI"     schemaLocation="../../wsdl/local/gw/api/test/TheirAPI.wsdl" /&gt;</pre>
PL-18297	Restored missing eachException methods on the Guidewire XML Modeler. Guidewire inadvertently removed these methods in a previous 7.x release.
<b>Infrastructure, Performance</b>	
CLM-17873	If you set a large number of days on the <b>Metrics &amp; Thresholds</b> screen of the <b>Administration</b> tab, new claims can either take many hours to complete, or they can hang and never advance to completion. The <b>Metrics &amp; Thresholds</b> screen now has a limit of 1,000 days as the maximum setting. If you try to set more than 1,000 days, you get an error message.  To configure this setting, open KeyMetrics.pcf in Studio, select the KeyMetrics widget in the PCF editor, and, on the Variables tab at the bottom of the screen, set the value of the MAX_METRIC_DAYS variable. Choose a value that limits metrics days to an optimal number that does not cause claim processing to delay or to fail.
<b>Integration (AB, PC, BC, ISO, Metro)</b>	
CLM-17191	This fix accommodates the addition of archiving to PolicyCenter 7.0.3 or integration with other policy systems that support policy archiving. It is now possible to include archived policies in the Policy Select search and the New Claim wizard policy search. To include archived policies in a policy search, you set the config.xml configuration parameter PolicySystemArchivingEnabled to true.  If any archived policies are returned by the search, there is a new column that shows that the policies are archived.  In addition, if you try to refresh a policy that has been archived in the policy system, you will now get an error.  <b>Note:</b> You must restore a policy in the policy system before you can use it in ClaimCenter.
CLM-17500	Contacts that are updated in ClaimCenter will now trigger automatic synchronization of other contacts if the config.xml setting InstantaneousContactAutoSync is true.
CLM-17540	When a Metropolitan Report Request is submitted with the same Report Type, Date of Loss, Insured Name, and Claim Number as a previously requested report, the new request will no longer get stuck in the Metropolitan Report workflow because of its duplicate status. This means that, for instance, metropolitan reports can now be received and attached to requests that are re-added after a metropolitan order timeout or workflow timeout has suspended the original request.
CLM-17623	On the Event Messaging screen of the Administration tab, you can now search for messages associated with a particular claim while viewing all messages for a destination. To see the search field, click any message destination on the Event Messaging screen. The screen for that destination shows a Claim # search field.
<b>History</b>	

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.
<b>Logging</b>	
PL-18234	<p>Improved logging of ConcurrentDataChangeException. Now, the log message shows both of the following:</p> <ul style="list-style-type: none"> <li>• The username of the current user, who received the ConcurrentDataChangeException</li> <li>• The previous user, who made the initial data modification</li> </ul> <p>The log writes the current user's name before the previous user's name in the message.</p>
<b>Messaging</b>	
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.
<b>Other - PL Services</b>	
PL-14011	<p>On the Server Tools tab, the Management Beans → CurrentUserSessions information now displays a sorted list of users. If a user is logged in more than once, ClaimCenter appends the number of sessions to the user name and puts the number in parentheses. For example:</p> <p>psmith(2),bsmith</p> <p>The intent is to preserve 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.</p>
<b>Performance</b>	
CLM-16748	Removed two ClaimCenter config.xml configuration parameters, DisableCBQTForClaimSearch and DisableCBQTForTeamGroupActivities. In versions of ClaimCenter prior to 7.0, there were problems with the cost-based optimizer in the version of Oracle then supported when performing certain claim search queries and team group activity page queries. These two parameters made it possible to disable certain features of the Oracle cost-based optimizer while executing these queries. Performance testing in ClaimCenter 7.0 has shown these configuration parameters are no longer necessary with the versions of Oracle that Guidewire now supports.
<b>Persistence</b>	
PL-17656	<p>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.</p> <p>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.</p>
<b>Policy, Reinsurance, Policy Refresh, LOB Configuration</b>	
CLM-16580	Attempting to refresh a policy or select a new policy on a claim with any unsaved changes on the claim will result in the error message, "You cannot refresh or select a policy on a claim with unsaved work". You must save any unsaved changes before refreshing a policy or selecting a new policy.
CLM-16899	Fixed an issue where a validation error would result from a ClaimCenter policy refresh after a vehicle was removed from the policy in PolicyCenter. The claim had a payment against an exposure related to the vehicle's incident. In this case, ClaimCenter now does not remove the vehicle from the policy.
CLM-17497	Policy.LossType does not work correctly and should not be used. Deprecated Policy.LossType in favor of the new property PotentialLossTypes, which returns all potential loss types that are compatible with the policy. Changed existing usages of Policy.LossType to use Claim.LossType instead.
<b>Plugins</b>	
PL-17162	Guidewire has added interface gw.plugin.SharedBundlePlugin. Any plugin that implements this interface shares the current bundle rather than creating a new one. For example, any plugin that implements IPreUpdateHandler now needs to add gw.plugin.SharedBundlePlugin to its list of implemented interfaces.
<b>Queries</b>	
PL-17676	Guidewire has deprecated the RawQuery property on gw.api.database.Query. Instead, create a gw.api.filters.StandardQueryFilter and use its filterQuery method or the IQueryBeanResult.addFilter method to apply the filter to a query.
PL-17719	

---

**Search - Core**

CLM-17085	<p>Changed the simple claim search screen to improve performance.</p> <p>Previously, if archiving was enabled, the simple claim search screen would check the archived claim information on every search. If ClaimCenter found any matching archived claims, it would show a link labeled <b>View archived claims (x)</b>, where x was the number of archived claims found.</p> <p>Getting the accurate count of matching archived claims took too much time, so now ClaimCenter does a simpler search to find if there are any matching archived claims, but does not try to get a count. If ClaimCenter finds any archived claims, the label for the link now just says, <b>View archived claims</b>. If there are no matching archived claims, the link is replaced by a message saying <b>No matching archived claims</b>.</p> <p>If archiving is not enabled, no extra search is done and there is no archived claims link or message.</p>
-----------	---

**Rules Infrastructure**

PL-17893	Guidewire has added a new <b>Info Page</b> to the <b>Server Tools</b> tab that shows the safe persisting order of the Preupdate rules. This page lists the order in which ClaimCenter runs the Preupdate rules for their root entities.
----------	---

**Security**

PL-10126	<p>Guidewire has added a new <b>Credentials</b> plugin to the base configuration. The <b>CredentialsPlugin</b> is similar to the <b>DBAuthenticationPlugin</b> and consists of the following three components:</p> <ul style="list-style-type: none"><li>• Interface <b>CredentialsPlugin</b> with the following sole method:     <pre>public UsernamePasswordPairBase retrieveUsernameAndPassword(String key)</pre></li><li>• Class <b>CredentialsUtil</b> with the following sole static method:     <pre>public static UsernamePasswordPairBase getCredentialsFromPlugin(String key)</pre></li><li>• Class <b>CredentialsPlugin.gs</b> provides a default implementation of the plugin in the base configuration.</li></ul> <p>Code can call the method <b>CredentialsUtil.getCredentialsFromPlugin</b>, 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 <b>Credentials.xml</b> file:</p> <ul style="list-style-type: none"><li>• If the code encounters the token <b>\${username}</b> as the value of the <b>username</b> value in the properties file or the token <b>\${password}</b> as the value of the <b>password</b> value in the properties file, then the code uses the <b>Credentials</b> plugin to obtain the real username and password.</li><li>• If the <b>Credentials</b> 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).</li></ul> <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> <p>In the base ClaimCenter configuration, the default implementation supports keys <b>ISO</b> and <b>Metro</b> and the code for <b>ISO</b> and <b>Metro</b> properties now uses the <b>Credentials</b> plugin.</p>
----------	--

The affected parameters in **ISO.properties** are:

- **ISO.CustLoginId**
- **ISO.CustPswd**

The affected parameters in **Metro.properties** are:

- **Metro.LoginId**
- **Metro.Pswd**

---

**SOAP APIs, Plugins, Rules - Core**

CLM-13888	The <b>SynchStateData</b> parameter has been removed from the <b>gw.webservice.cc.claim.IClaimAPI</b> methods <b>migrateClaim</b> , <b>addFNOL</b> , and <b>addExposure</b> because it is no longer used in the implementation. <b>SynchStateData</b> is also deprecated.
CLM-16222	An issue where related contacts were being erroneously deleted in ContactManager when updated in ClaimCenter has been fixed.
CLM-16365	You are now prevented from adding a date of birth (DOB) that occurs in the future in the New Claim wizard. The fix was to add UI-level validation preventing a DOB in the future in the DOB input fields in <b>AddressBookAdditionalInfoInputSet.Person.pcf</b> and <b>AdditionalInfoInputSet.Person.pcf</b> .
CLM-16222	An issue where related contacts were being erroneously deleted in ContactManager when updated in ClaimCenter has been fixed.

---

**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	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.  Guidewire now displays a progress dialog to indicate the delay.
PL-18110	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.  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.
PL-18112	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.  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.
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:  <pre>uses java.util.HashSet uses gw.lang.reflect.IType  var o = new HashSet&lt;IType&gt;() {A, B, C, ...} return o.contains(typeof(...))</pre>
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"> <li>• It detects lost connections very quickly if debugging as there is immediate feedback from the debug API.</li> <li>• 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.</li> <li>• If Studio detects the server has restarted, Studio reconnects automatically.</li> </ul>
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 ClaimCenter stored the initial value of a newly created script parameter in an incorrect location, thus invalidating the application installation checksum. ClaimCenter correctly stored subsequent entries in the correct location.

**Studio IDE - PCF Editor**

PL-11489	Guidewire has added a new PCF Verification option ( <b>Tools → Options → Verification Options</b> ) that you use to enable/disable whether ClaimCenter 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>ClaimCenter 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"><li>• 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.</li><li>• 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.</li></ul> 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.
<b>Utilities</b>	
PL-17920	Guidewire has removed the following JAR files from the application build: <ul style="list-style-type: none"><li>• activation.jar</li><li>• activation-1.1.1.jar</li></ul>
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"><li>• mailapi.jar</li><li>• smtp.jar</li></ul>
<b>Web - Other</b>	
PL-17861	Added a <code>disablePostOnEnter</code> attribute to the Input Group widget.
<b>Web - UI/Runtime</b>	
PL-18233	Fixed an issue that failed to highlight in yellow a problematic <code>RadioButtonCell</code> field even if the data value for the <code>RadioButtonCell</code> contained an error.
<b>Web Services - WSI</b>	
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 <code>security-configuration</code> tag of WebLogic's <code>config.xml</code> :  <code>&lt;enforce-valid-basic-auth-credentials&gt;false&lt;/enforce-valid-basic-auth-credentials&gt;</code>
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 ( <code>wait</code> , <code>notify</code> , 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.
<b>Workflow</b>	
PL-9668	Added new configuration parameter <code>WorkflowLogDebug</code> , which takes a Boolean value: <ul style="list-style-type: none"><li>• If set to <code>true</code>, ClaimCenter outputs the ordinary verbose system workflow log messages from the Guidewire server to the workflow log.</li><li>• If set to <code>false</code>, ClaimCenter does not output any of the ordinary system messages.</li></ul> 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 <code>cc_workflowlog</code> table, Guidewire recommends that you set this parameter to <code>false</code> .

PL-17556	It is now possible to run the following writer batch processes from either the <b>Batch Process Info</b> screen or the <b>Work Queue Info</b> screen on the <b>Server Tools</b> tab.: <ul style="list-style-type: none"><li>• Activity Escalation</li><li>• Group Exception</li><li>• User Exception</li></ul>
<b>XML Element (and XSD types)</b>	
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 ClaimCenter.

- “ClaimCenter Known Issues” on page 286
- “Studio/Platform Issues” on page 287

**Note:** For maintenance releases, Guidewire often defers 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.

### ClaimCenter Known Issues

#### New Claim wizard not saving selected vehicle in some cases (CLM-13973)

**Issue** – The second step of the New Claim wizard is not saving your vehicle selection when you leave the screen and then come back. For example, you choose a vehicle and then click the **Policy Details** link, and when you click **Basic Info** on the left to return to this screen, the vehicle is no longer selected.

**Workaround** – Reselect the vehicle when you return to the wizard screen.

#### Upgrade from ClaimCenter 6 to ClaimCenter 7 does not work for SQLServer if MigrateToLargeIDsAndDatetime2 is set to true (CLM-17739)

**Issue** – During upgrade with SQLServer as the database, the `cc_tmppagglimitrpt` table fails to be upgraded if it is not empty and `MigrateToLargeIDsAndDatetime2` is `true`.

**Workaround** – If there is data in the table `cc_tmppagglimitrpt`, truncate the table before beginning the upgrade by running the following SQL command:

```
DELETE FROM cc_tmppagglimitrpt
```

#### ABContactPlugin now requires ContactManager 7.0.1 (CLM-17990)

**Issue** – The class `gw.plugin.addressbook.ab700.PluginRetrieveOperation`, which is used by ABContactPlugin, has a method that requires the ABContactAPI web service in ContactManager 7.0.1. This method, `retrieveReplacementContact`, will cause errors if your ClaimCenter application continues to be integrated with ContactManager 7.0.0. Guidewire recommends that you upgrade to ContactManager 7.0.1. If you cannot upgrade to ContactManager 7.0.1, see the workaround for this issue.

**Workaround** – Copy `ClaimCenter/modules/cc/gsrc/gw/plugin/addressbook/ab700/PluginRetrieveOperation.gs` to `ClaimCenter/modules/configuration/gsrc/gw/plugin/addressbook/ab700/PluginRetrieveOperation.gs`. Edit the copied file and remove the annotation `@ReadOnly`, the `retrieveReplacementContact` method, and all calls to the `retrieveReplacementContact` method. Then save the file.

## When policy refresh does not find policy, error message is not useful (CLM-18170)

**Issue** – If ClaimCenter is integrated with PolicyCenter and you refresh a policy, but PolicyCenter cannot find the policy, the error message you see is “Error: null”. What you want to see is an error message about the policy’s not being found.

**Workaround** – Open ClaimCenter Studio, edit `gw.plugin.pcintegration.PC702.PolicySearchPCPlugin.retrievePolicy`, and find the following line of code:

```
if(resultSet.Result.Status.equals(PolicyStatus.TC_ARCHIVED)) {
```

Change this line of code to:

```
if(PolicyStatus.TC_ARCHIVED.equals(resultSet.Result.Status)) {
```

## Address Fields shown incorrectly in New Claim wizard Step 3 with duplicate claim (CLM-18187)

**Issue** – When you are working on a duplicate claim in the New Claim wizard, in the **Loss Details** step, the screen is missing a number of address fields, including **Address 1**, **Address 2**, **City**, **County**, **State**, **ZIP Code**, and **Country**.

**Workaround** – Refresh the page, ignoring the warning about possible loss of data. Everything you entered before the **Loss Details** wizard step is saved. After refreshing the page, the address fields become visible, and you can continue with this step.

## Studio/Platform 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:

```
ClaimCenter/modules/cc/config/registry/compatibility-xsd.xml
```

To:

```
ClaimCenter/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.ui.RunTestCommand.buildTestSuite(RunTestCommand.java:109
...
...
```

**Workaround** – Run the following command before you run a test:

```
.../bin/gwcc dev-deploy
```

## Problem with *regen-java-api* command and JAR files (PL-16351)

**Issue** – If you run the *ClaimCenter/bin/gwpc regen-java-api* command, ClaimCenter creates a *ClaimCenter/java-api/lib* directory and puts JAR files for Java development in this directory. However, ClaimCenter 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:

```
ClaimCenter/admin/lib
```

Copy them into the following directory:

```
ClaimCenter/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, ClaimCenter throws *ParseResultsException*. This is the intended behavior.

**Workaround** – Restart the workflow engine. To do so:

1. Log into ClaimCenter 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:

```
gwcc 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">
```

## When entering a money amount, thousands and decimal separators that do not match the current locale are ignored instead of generating an error (PL-14348)

**Issue** – When entering a money amount, if you use a separator character that does not match the separator character for the current locale, that character will be ignored rather than causing an error. This behavior can lead to the number's being interpreted as a different amount than you intend. For example, in a French locale, the decimal separator is a comma, and the thousands separator is a space. In ClaimCenter 7.0.0 and 7.0.1, if you enter 1002.50, the period character will be ignored, and the amount will be interpreted as if you had entered 100250, one hundred times greater than the amount you intended to enter. In ClaimCenter 6.0.x, this entry would have generated an error message.

**Workaround** – When entering money amounts, be sure to use the correct decimal separator for the current locale and check that the resulting amount was interpreted correctly. Guidewire is aware of this issue.

---

## chapter 30

# Guidewire ClaimCenter 7.0.0 Release Notes

## Release 7.0.0

**Document Revision:** 23-June-2016

Copyright © 2001-2016 Guidewire Software, Inc.

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, Guidewire Spotlight, Gosu, Adapt and succeed, 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.

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.

## Overview

These release notes contain the following sections:

- “Release Information” on page 292
- “Installing This Release” on page 292
- “Changes in This Release Provided in Upgrade Diff Report” on page 292

- “Known Issues and Limitations” on page 293

## Release Information

These release notes apply only to this release of Guidewire ClaimCenter.

### Version Number

This release of Guidewire ClaimCenter is 7.0.0.

For Standard Reporting, ClaimCenter 7.0.0 supports IBM Cognos Business Intelligence 10.1 as the report server.

### Support

This document includes important information about the current release of Guidewire ClaimCenter. For assistance with this software release, contact Guidewire Customer Support:

- At the Guidewire Resource Center – <http://guidewire.custhelp.com>
- By email – [support@guidewire.com](mailto:support@guidewire.com)
- By phone – +1-650-356-4955

## Installing This Release

Refer to the following:

- The *ClaimCenter Installation Guide* for general installation information
- The *Guidewire Contact Management Guide* to install ContactManager
- Prior ClaimCenter release notes for any versions that you have skipped

### Upgrading Coverage Terms (CLM-15711, CLM-15906)

The data model for the CovTerm entity on a Policy has changed in ClaimCenter 7.0. As part of upgrading from a prior version, you need to either keep the old data model and its associated PCF files, or use the new ClaimCenter 7.0 data model and its PCF files. This affects both the database and configuration upgrade.

AggregationModel, RestrictionModel, ValueType, and Type fields were deleted in the CovTerm entity. These were all free-form text fields in prior versions of ClaimCenter. New fields have been added in ClaimCenter 7.0 to store this data in a type-safe way that is more consistent with PolicyCenter: CovTerm.CovTermPattern, CovTerm.ModelAggregation, CovTerm.ModelRestriction, and NumericCovTerm.CovTermModelVal.

The Policy → Locations page has been redesigned to accommodate this change and other data model changes. To handle the coverage terms changes, the ClaimPolicyCovTermsLV.pcf replaces CovTermLV.pcf used in prior versions.

If you have data in these coverage term fields and wish to keep this data, please contact Guidewire Support for instructions.

## 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 ClaimCenter:

- “ClaimCenter Known Issues” on page 293
- “Platform/Studio Known Issues” on page 296

### ClaimCenter Known Issues

#### Reinsurance Issues

**Issue** – In addition to the listed known issues, there are also several known issues with the new Reinsurance capabilities in ClaimCenter 7.0.0.

**Workaround** – If you plan to begin implementing reinsurance management in ClaimCenter 7.0.0, please contact Support for help in devising an implementation strategy that will ensure a smoother upgrade process.

#### Prematurely deprecated methods on FinancialCalculationUtil (CLM-17710)

**Issue** – Methods on `FinancialsCalculationUtil` were deprecated prematurely. Methods such as `FinancialsCalculationUtil.getAvailableReserves()` which return a type of `FinancialsCalculation`, will have their deprecation removed in CC 7.0.1.

**Workaround** – You do not need to change code which uses these deprecated methods.

#### ClaimCenter policy search supports searching on city, state, zip code, and country, but PolicyCenter policy search does not (CLM-17669)

**Issue** – ClaimCenter shows city, state, zip, and country on the **Policy Search** page, but state, zip, and country are ignored by the PolicyCenter policy search plugin. This plugin is located at `gw.plugin.pcintegration.pc700.PolicySearchPCPlugin`.

**Workaround** – The `PolicySearchPCPlugin` in ClaimCenter can be modified to work with PolicyCenter 7.0.2. After the following modification, the `PolicySearchPCPlugin` will return city, state, zip and country from PolicyCenter to ClaimCenter.

1. In the `ClaimCenter` directory, double-click on `studio.bat` to start Guidewire Studio.
2. In the **Resources** frame, expand the **RPC-Encoded Web Services** subtree.
3. Select `pcintegrationPC700`.
4. Click the **Refresh** button next to the **URL** field to refresh the wsdl.
5. In the **Resources** frame, open `Classes → gw → plugin → pcintegration → pc700 → PolicySearchConverter`.
6. Locate the following line in this file:  

```
pcCriteria.State = ccCriteria.InsuredAddress.State.Code
```
7. Replace this line with the following lines:  

```
pcCriteria.PrimaryInsuredCity = ccCriteria.InsuredAddress.City  
pcCriteria.PrimaryInsuredState = ccCriteria.InsuredAddress.State.Code  
pcCriteria.PrimaryInsuredPostalCode = ccCriteria.InsuredAddress.PostalCode  
pcCriteria.PrimaryInsuredCountry = ccCriteria.InsuredAddress.Country.Code
```
8. Save the file.
9. Restart the ClaimCenter server.

## Adding or editing an existing contact that has no ContactTag markers in the New Claim Wizard prevents the wizard from moving to completion (CLM-17594)

**Issue** – In the New Claim Wizard, a `ConcurrentDataChangeEvent` is issued after you click **Finish** when all of the following conditions are true:

- You add a contact to the claim from the **Address Book**.
- This contact has not been tagged with the **ContactTag** markers that ClaimCenter normally adds to the contact. This happens when:
  - The contact has never been on a claim and ContactManager has not set any **ContactTag** markers for it.
  - The contact is a related contact on the claim. Related contacts are retrieved from ContactManager as stub contacts without tags. For instance, an insured's employer is automatically set on a claim, but it has no tags.
- You edit the contact that has no tags.

When the New Claim Wizard encounters this condition, it issues the following error message: `The object you are trying to update was 'contact name', and it was changed by System User at time.`

**Workaround** – Add or edit the contact from the **Parties Involved** page on the claim after you have successfully completed the wizard.

## Concurrent restore of archived claim by second user needs to give better error message (CLM-16642)

**Issue** – ClaimCenter does not prevent a second user from attempting to restore an archived claim around the same time that another user is restoring the same claim. ClaimCenter can cache the archived state of the claim if they both enter the **Archived Claim** screen around the same time. ClaimCenter does not guarantee mutual exclusion during the restore process. If this happens, then one user's restore will succeed and the other user will receive an exception.

**Workaround** – Guidewire is aware of this issue. Guidewire recommends that your implementation of the `IArchiveSource` plugin check if the claim has already been restored. If so, throw a `DisplayableException` instead of returning the archived claim document to be restored.

## Changing loss date does not retrieve modified policy (CLM-15400)

**Issue** – ClaimCenter is not retrieving the new policy period in the FNOL wizard when integrated with PolicyCenter. This only occurs if you select the policy with a date and then you go back in the wizard and attempt to change the date.

**Workaround** – Cancel out of the FNOL wizard and then restart it. Enter the correct loss date from the beginning.

## Manual exchange rate printing is incorrect (CLM-14903)

**Issue** – If you print any financials pages in multicurrency mode, the manual exchange rate print out is incorrect. The issue is caused by incorrect printing when using the `format` attribute on a PCF widget.

**Workaround** – You must first correct the `TransactionExchangeRateInputSet.pcf` file. The input in question is:

```
<TextInput
  editable="transaction.OverrideTransToClaimExchangeRate"
  format="var rate = transaction.TransToClaimExchangeRate; return &quot;1 &quot; +
  rate.BaseCurrency.DisplayName + &quot; = #.##### &quot; + rate.PriceCurrency.DisplayName;" 
  formatType="exactNumber"
  id="Transaction_ExchangeRate"
  label="displaykey.Web.Claim.ExchangeRateInputSet.ExchangeRate"
  numCols="8"
  postOnChange="true"
  value="transaction.TransToClaimExchangeRateRate"/>
```

Do the following:

1. First, convert this to a generic Input widget, without the `format` attribute.

2. Next, add a custom entity display name for the ExchangeRate entity in Studio (path is configuration → Entity Names).
3. You can customize how you would like the exchange rate to show.

### Dashboard statistics batch process can perform slowly (CLM-14707)

**Issue** – The dashboard statistics batch process can perform slowly.

**Workaround** – Create a histogram on the createtime column on the cc\_transaction table.

### Duplicate descriptions for image/jpg mime types in the configuration file (CLM-14188)

**Issue** – In the config.xml file, there are two entries under MIME types for jpeg file types:

- pjpeg
- jpeg

This causes a problem as the descriptions of both are identical. If you upload a standard jpeg and chooses the wrong jpeg MIME type, an error occurs.

**Workaround** – Change the description of the pjpeg so that duplicate entries do not appear in the MIME type drop-down menu in Guidewire Studio.

### Additional calls to a policy administration system are made when a duplicate claim check runs (CLM-14078)

**Issue** – While creating a new claim, you select a policy and click **Next**. A call is made to the policy administration system (PAS). The policy adapter retrieves all the policy data, coverages, endorsements, contacts, and so forth and sends this information to ClaimCenter. If you receive the **Duplicate Claim** warning, you must click **Close** on the warning to proceed with claim set-up. However, when you click **Close**, you must click **Next** again which results in another (unnecessary) call to the PAS.

**Workaround** – The **Next** button is blocked by the `DisplayableException` thrown in the nested function `checkForDuplicates` in the `NewClaimWizard.pcf` file as seen in the following example:

```
function checkForDuplicates() {
    if (Wizard.checkForNewDuplicateClaims()) {
        NewClaimDuplicatesWorksheet.goInWorkspace(Wizard);
        throw new util.DisplayableException(displaykey.Java.NewClaimWizard.DuplicateClaimWarning);
    }
}
```

This is executed on the Main Contacts wizard step:

```
<WizardStep id="MainContacts"
    label="displaykey.Wizard.NewClaimWizard.NewClaimMainContacts.Label"
    ...
    onExit="checkForDuplicates()"
    ...
```

To allow **Next** to execute, you can create a new function such as:

```
function checkForDuplicatesNoWarning() {
    if (Wizard.checkForNewDuplicateClaims()) {
        NewClaimDuplicatesWorksheet.goInWorkspace(Wizard)
    }
}
```

Have the `onExit` function of the Main Contacts wizard step call:

```
onExit="checkForDuplicatesNoWarning()
```

Now, you can go to the next step. However, you are still warned of duplicate claims encountered. The **Next** button proceeds to the next step in the wizard, thus avoiding extra queries to the external policy system.

## FNOL wizard not saving selected vehicle (CLM-13973)

**Issue** – The second step of the FNOL wizard does not save the selected vehicle when viewing the **Policy Details** screen and then returning to the **Basic Info** screen.

**Workaround** – In the `FNOLWizard_BasicInfoScreen.default.pcf` file, there is an `InputGroup` with ID `InsuredVehicleInputGroup`. Add the `onToggle` attribute:

```
onToggle="!InsuredVehicle.Checked = !InsuredVehicle.Checked; Wizard.synchronizeVehicleIncidents(Claim)"
```

## *IClaimAPI depends on SynchStateData, which uses deprecated methods (CLM-13888)*

**Issue** – `IClaimAPI` depends on `SynchStateData`, which in turn uses deprecated functions like `setMessageSinkID` and `getMessageSinkID`. For this reason, the `SynchStateData` parameter is ignored.

**Workaround** – When calling `IClaimAPI`, pass in null for the `SynchStateData` parameter.

## Incorrect additivity statements in the default logging.properties file (CLM-13617)

**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
```

The statements for API, Database, Messaging, `Messaging.ISO`, `Plugin.ClaimNumGenLog`, `Plugin.IPolicySearchAdapter`, and `Plugin.IContactSearchAdapter` are also incorrect and need to be changed.

## Japanese layout issue in Advanced Search screen (CLM-13594)

**Issue** – When the user interface is localized to the Japanese language, the source label at the top left-hand side of the **Advanced Search** screen goes down the left side instead of across.

**Workaround** – Edit the `RangeInput` widget on `ClaimSearchDV` and set `labelAbove="true"`.

## ClaimCenter renders some user interface screens in Japanese incorrectly (CLM-13508)

**Issue** – When the user interface is localized to the Japanese language, the **Open Recovery Reserves** screen, and the **Cost Type** and **Cost Category** on the **New Recovery Reserve** multicurrency popup are incorrectly rendered.

**Workaround** – Set the `wrapLabel` property to `false` and set the `width` property to 10 for these cells.

## Platform/Studio 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.

## Bing Geocode plugin implementation not visible in Studio (PL-17675)

**Issue** – The Bing Geocode plugin implementation (`BingMapsPlugin.gs`) is not visible in Studio.

**Workaround** – To access the Gosu source code for this plugin, access it from the file system within your Guidewire application directory at the following location:

```
...modules/p1/gsrc/gw/plugin/geocode/impl
```

**WARNING** Do not modify the file in this folder as that invalidates the application checksum. To modify `BingMapsPlugin.gs`, first copy it to `.../configuration/gsrc/gw/plugin/geocode/impl`.

## Problem with regen-java-api command and JAR files (PL-16351)

**Issue** – If you run the `ClaimCenter/bin/gwpc regen-java-api` command, ClaimCenter creates a `ClaimCenter/java-api/lib` directory and puts JAR files for Java development in this directory. However, ClaimCenter 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:

```
ClaimCenter/admin/lib
```

into the following directory:

```
ClaimCenter/java-api/lib
```

## 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.

## 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.

## 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.

## 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.

## 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.

## 'ListDetailPanel' throws exception (PL-10316)

**Issue** – It is possible for ClaimCenter 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.

## 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:**

*PolicyCenter/modules/p1/config/registry/compatibility-xsd.xml*

To:

*PolicyCenter/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.

## 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.